

'68'

\$2.50 USA

Australia
Singapore

A \$ 4.00
S \$ 8.00
Malaysia

New Zealand NZ \$ 4.00
Hong Kong H \$20.00
M \$ 8.00

MICRO JOURNAL

VOLUME II ISSUE 3 • Devoted to the 68XX User • March 1980
"Small Computers Doing Big Things."

SERVING THE 6800 USERS WORLDWIDE

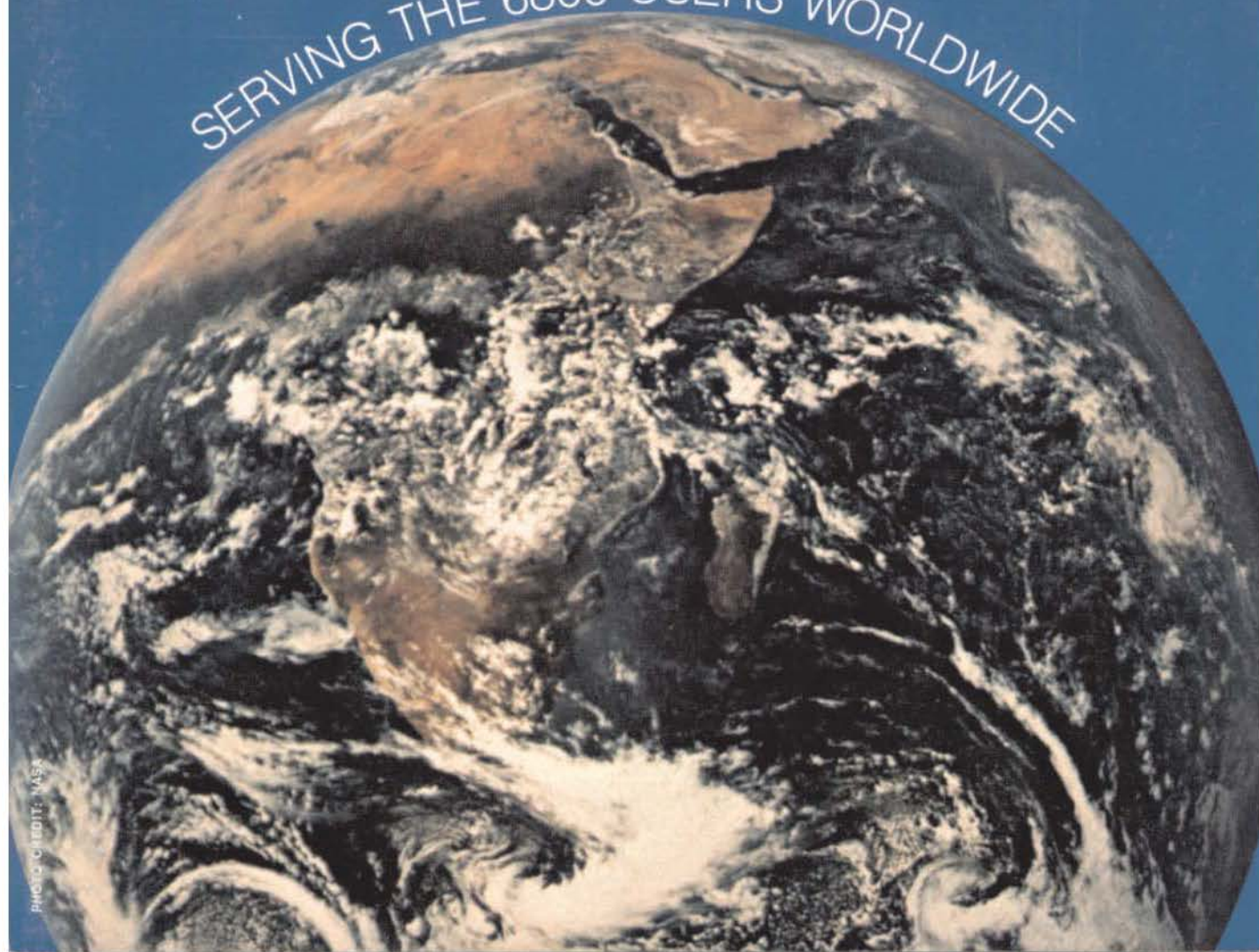


PHOTO CREDIT: NASA



SYSTEMS - SOLUTIONS

If you have a problem that can be solved by a computer—we have a systems solution.

- Two central processors with maximum RAM capacities of 56K and 384 K bytes
- Three types of disk drives with capacities of 175K, 1.2M and 16M bytes
- Two dot matrix printers with 80 and 132 line capacity
- A Selectric typewriter interface and a daisy wheel printer

Match these to your exact need, add one or more of our intelligent terminals and put together a system from one source with guaranteed compatibility in both software and hardware.

Southwest Technical Products systems give you unmatched power, speed and versatility. They are packaged in custom designed woodgrain finished cabinets. Factory service and support on the entire system and local service is available in many cities.



SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216

(512) 344-0241

The Source for 6800/6809 Software

Technical Systems Consultants, Inc. is The Source for your 6800/6809 systems software needs. From FLEX™, the standard disk operating system of the 680X family, to Sort/Merge, your systems requirements can be filled with the highest quality software in the industry. Nowhere else can you find such variety from a single source. Here are some of the most popular:

Program Name	6800	6809
FLEX for SWTPc	\$90	\$90
FLEX for SSB	90	90
Extended BASIC	100	100
Extended BASIC Precompiler	50	50
BASIC	65	65
BASIC Precompiler	40	40
FLEX Sort/Merge	75	75
Text Editing System	40	35
Assembler	40	40
Text Processing System	60	N/A
Debug Package	55	75
FLEX Utilities	100	60

These packages are available on either 8" or 5" soft-sectored FLEX diskettes (5" 6800 is FLEX 2.0). Price includes user's manual and object code diskette. Certain programs are available on cassette. Contact Technical Systems Consultants for pricing. All orders should include 3 percent for postage and handling (8 percent on foreign orders). Master Charge and Visa are welcome.

*FLEX is a trademark of Technical Systems Consultants, Inc.



**technical systems
consultants, inc.**

Box 2574, West Lafayette, IN 47906
(317) 463-2502

'68'

Portions of the text of '68' Micro Journal set using the following
6800/2, DMAF1 and CT-82
Southwest Technical Products Corp.
219 W Rhapsody
San Antonio, TX 78216

Editor, Word Processor and Sort/Merge
Technical Systems Consultants, Inc.
Box 2574
W. Lafayette, IN 47906 *MINIFLEX & FLEX REG.*
Technical Systems Consultants, Inc.

Selectric I/O
World Wide Electronics, Inc.
130 Northwestern Blvd.
Nashua, NH 03060

Publisher/Editor
Don Williams Sr.
Executive Editor
Larry E. Williams

Contributing Editors

Mickey Ferguson
Dennis Womack
Dr. Jack Bryant
Dr. Chuck Adams
Dr. Theo Elbert
Dr. Jeffery Brownstein
Dale Puckett
Howard Berenbon
Russell Gore
T. Jackson - Japan

Subscriptions and Office Manager
Joyce Williams

Typography and Color Separations
Williams Company, Inc.
Chattanooga, TN 37421

* CONTENTS *

Multi-User Ed System...8	Gerhold & Kherlaty
Bookkeeping-Part 2.....10	Stock
Dumpfile and Dumpcmds...15	Pigford
SWTPC to PDP-11 PDP-11 to SWTPC.....22	
Trap (FLEX™).....23	Johnson
TSC Random Files.....23	Schreier
Tape Labeling.....26	Looney
Cassette BASIC K (AAA)...28	Reader Review
Letters, New Prod, Etc.28	
Classified.....29	
West Coast Comp Fair...29	

MICRO JOURNAL

Send All Correspondence To:

'68' Micro Journal
3018 Hamill Rd.
PO Box 849
Hixson, Tennessee 37343

— Phone —
Office: 615-870-1993
Plant: 615-892-7544
Copyright © 1980

'68' Micro Journal is published 12 times a year by '68' Micro Journal, 6131 Airways Blvd., Chattanooga, TN 37421. Second Class postage paid at Chattanooga, TN. Postmaster: Send Form 3579 to '68' Micro Journal, PO Box 849, Hixson, TN 37343.

1-Year \$14.50 2 Years \$26.00 3 Years \$36.50

—ITEMS SUBMITTED FOR PUBLICATION—

(Letters to the Editor for Publication) All letters to the Editor should be substantiated by facts. Opinions should be indicated as such. All letters must be signed. We are interested in receiving letters that will benefit or alert our readers. Praise as well as gripes is always good subject matter. Your name may be withheld upon request. If you have had a good experience with a 6800 vendor please put it in a letter. If the experience was bad put that in a letter also. Remember, if you tell us who they are then it is only fair that your name not be withheld. This means that all letters published, of a critical nature, cannot have a name withheld. We will attempt to publish 'verbatim' letters that are composed using 'good taste.' We reserve the right to define (for '68' Micro) what constitutes 'good taste.'

(Articles and items submitted for publication) Please, always include your full name, address, and telephone number. Date and number all sheets. TYPE them if you can, poorly handwritten copy is sometimes the difference between go, no-go. All items should be on 8X11 inch, white paper. Most all art work will be reproduced photographically, this includes all listings, diagrams and other non-text material. All typewritten copy should be done with a NEW RIBBON. All hand drawn art should be black on white paper. Please no hand written code items over 50 bytes. Neatly typed copy will be directly reproduced. Column width should be 3¼ inches.

(Advertising) Any Classified: Maximum 20 words. All single letters and/or numbers will be considered one (1) word. No Commercial or Business Type Classified advertising. Classified ads will be published in our standard format. Classified ads \$7.50 one time run, paid in advance.

Commercial and/or Business advertisers please write or phone for current rate sheet and publication lag time.



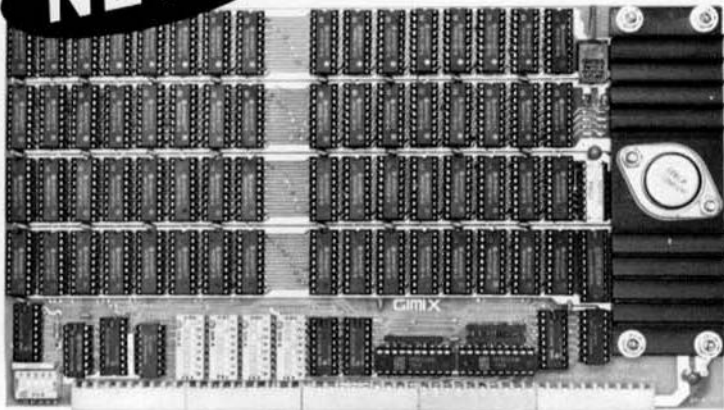
WANT TO BE KING OF THE HILL?

TREAT YOURSELF ROYALLY WITH GIMIX UNIQUE AND INCOMPARABLE
BOARDS AND SYSTEMS... DIP-switch Versatility for use with
both SS50 (6800) and SS50C (6809) Systems (SWTP. etc.)

THE FIRST AND ONLY 32K STATIC RAM BOARD...

Designed for use with: ★ Existing SS50 Systems
★ SS50C Extended Address Systems

NEW!



FEATURES:

- Decoding for 4 Extended Address Lines (allows memory decoding up to 1 megabyte)
- DIP-switch to set extended addressing or disable it
- 4 separate 8K blocks, addressable to any 8K boundary by DIP-switch
- Each 8K block may be individually disabled
- Write protect either of two 16K sections
- Low power consumption — uses 2114L low power RAMS — (2 amps typical for 32K)
- Fully Socketed
- Gold Bus Connectors

Assembled, Burned In and Tested at 2MHz.

16K \$328.12 24K \$438.14 32K \$548.15

16K and 24K Versions are socketed for 32K and require only additional 2114S for expansion.

FACTORY PRIME STATIC RAMS

2114L 450 ns \$5.90 200 ns \$6.90

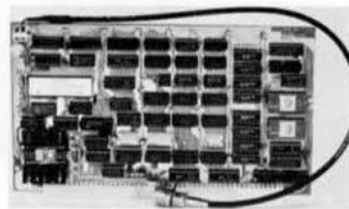
4044 450 ns \$5.90 250 ns \$6.90

ADD \$5.00 HANDLING ON ORDERS UNDER \$200.00

THE UNIQUE GIMIX 80 x 24 VIDEO BOARD

- ★ Upper and Lower Case with Descenders ★ Hardware Scrolling
 - ★ Contiguous 8x10 Character Cells ★ X-Y Addressable Hardware Cursor
- It is the **ONLY** Video Board that gives you:

- A user programmable RAM character generator. Custom character sets, up to 128 characters each, can be stored and loaded into the board **under software control**, from disk, tape, etc.
- The ability to choose, **under software control**, 256 displayable characters from 384 available in the 3 on board (2 EPROM and 1 RAM) character generators.
- The ability to divide the 256 displayable characters into 8 groups, according to **both** ASCII Code and bit 8; lets your program determine how each group is displayed. (Which character generator to use, and whether it will be normal or Inverse video, full or reduced intensity or a combination of these.)
- GHOSTability: to place multiple boards at the same address and access them individually without affecting the display of the other boards.
- The ability to control all these features, on the fly, through software.



Fully decoded, occupies only 2K of address space.

Fully socketed — Gold bus connectors. Assembled, Burned in, and Tested at 2MHz.

Deluxe Version with RAM
Character Generator \$458.76
Without RAM Character
Generator \$398.74

Also Available...
64 or 32 x 16 Video Board ... \$198.71

Phone, write, or see your dealer for details and prices on our broad range of Boards and Systems for the SS50/SS50C bus and our AC Power Control Products for all computers.

GIMIX inc.

1337 WEST 37th PLACE, CHICAGO, IL 60609
(312) 927-5510 • TWX 910-221-4055

The Company that delivers.
Quality Electronic products since 1975.

GIMIX® and GHOST®
are Registered Trademarks of GIMIX INC.

THE CLASSY CHASSIS

With Baud Rate Generator
on Mother Board \$828.19

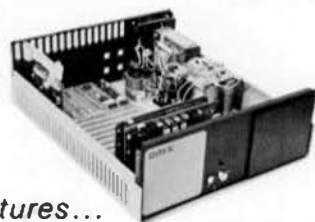
32K SYSTEM Incomparable Features...

at a Comparable Price! \$1,594.59

Includes: Chassis, 6800 CPU, 32K RAM Board, Choice of I/O Card.

16K Version of above \$1,374.49

- Ferro-Resonant Power Supply (+8V at 25 Amps, + and - 16V at 5 Amps each.)
- 6800/6809 Mother Board, has fifteen 50 pin plus 8 DIP-switch addressable 30 pin slots, fully decoded to 4, 8 or 16 addresses — Gold Plated Pins.
- Heavy Weight aluminum cabinet with fan and provisions for 1 or 2, 5 inch disk drives.



SEE GHOST AD PAGE 43



GIMIX inc.

1337 WEST 37th PLACE • CHICAGO, ILLINOIS 60609 • (312) 927-5510 • TWX 910-221-4055

We have a wide variety of 8080 bus items available. They are compatible with existing SWTP, MSI, and other manufacturer's products and the DIP-switch features on GIMIX systems allow use of a broad range of existing software. The below items are normally in stock except for systems which are customized to your needs. If you don't see what you want, please ask. We believe in service. All our boards and systems are assembled, burnt-in, and tested.

HANDLING CHARGES
U.S. On Orders Under \$200.00
Add \$5.00

32K 6800 SYSTEMS: Include 6800 CPU Board, 32K RAM Board, Mainframe Cabinet, Mother Board, Power Supply, Fan, Choice of One 30 Pin I/O Card and Cable, Choice of Two Filler Plates or Two Disk Regulator Cards, Binder, Documentation, and Manuals		\$1,594.59	SOFTWARE		
16K 6800 SYSTEMS: As Above, With 16K Version of 32K RAM Board		1,374.49	GMXBUG MONITORS:		
For 6800 CPU with 6840 Timer Option		Add 64.00	2K VERSION 1 — ON 2 2708s		68.54
VIDEO BASED SYSTEMS: When Ordered With Either of the Above.			3K VERSION 2 — ON 3 2708s		88.55
Video Based Systems Require Only A Parallel Keyboard and A Video Monitor and Are Faster, Lower In Cost, and Give You More Versatility and Features For Your Money Than Most Terminals.			3K VERSION 3 — ON 3 2708s		98.56
64 x 16 VIDEO BOARD PLUS GMXBUG V.2.0		Add 200.00	VERSION 1 MANUAL ONLY WITH SOURCE LISTINGS — NO PROMS		28.51
80 x 24 VIDEO BOARD (Without RAM Set) PLUS GMXBUG V.3.0		Add 400.00	VERSION 2 MANUAL ONLY WITH SOURCE LISTINGS — NO PROMS		28.52
80 x 24 VIDEO BOARD (Full Features) PLUS GMXBUG V.3.0 PLUS THE MANUAL AND DISK FOR PROGRAMMING USER DEFINED CHARACTER SETS		Add 450.00	VERSION 3 MANUAL ONLY WITH SOURCE LISTINGS — NO PROMS		38.53
12 INCH VIDEO MONITOR (60Hz Only), Keyboard and Enclosure, Cables and Connectors		350.00	PROGRAM TO CREATE USER DESIGNED CHARACTERS — MANUAL & DISK		20.61
6800/6809 MAINFRAME (Includes Chassis, Power Supply, Switches, Fan and Mother Board Plus Baud Rate Generator on Mother Board)		828.19	POWER CONTROL BOARDS		
6800/6809 MOTHER BOARD ONLY		294.09	Contact GIMIX for Pricing on Complete Turnkey GHOST Systems with Real Time Software.		848.08
DISK REGULATOR BOARD		18.91	RELAY DRIVER PACKAGE Assembled and Tested With 31 Relays, Bracket and Transformer		448.86
FILLER PLATES		14.92	RELAY DRIVER BOARD ONLY		38.21
6800 CPU BOARD — WITHOUT TIMER		224.03	BRACKET		14.24
6800 CPU BOARD WITH 6840 TIMER PACKAGE		288.06	TRANSFORMER		11.50
MEMORY BOARDS			G.E. RR8 20 AMP RELAY		348.85
32K STATIC RAM, SOCKETED, WITH EXTENDED ADDRESSING		548.15	OPTO ISOLATED INPUT BOARD (34 SWITCH CLOSURES)		98.83
24K VERSION OF ABOVE (SOCKETED FOR 32K)		438.14	16-BUTTON REMOTE 2-WIRE KEYBOARD SYSTEM KEYBOARD		248.84
16K VERSION OF ABOVE (SOCKETED FOR 32K)		328.12	35 BUTTON SERIAL KEYBOARDS — Wood Laminate, Black, White or Walnut		118.82
16K GHOSTable STATIC RAM		68.16	35 BUTTON SERIAL KEYBOARDS — Clear or Smoke Acrylic		128.82
8K 2708 EPROM BOARD		98.34	FACTORY PRIME CHIPS		
I.O. BOARDS			4044s		450 ns 5.90
30 PIN SINGLE PORT SERIAL		88.41	4044s		250 ns 6.90
30 PIN 2 PORT PARALLEL		88.42	2114Ls		450 ns 5.90
50 PIN 8 PORT SERIAL (Available Feb. '80)		288.40	2114s		200 ns 6.90
50 PIN 8 PORT SERIAL WITH BAUD RATE GENERATORS		318.66	2708s		450 ns 8.90
50 PIN 8 PORT PARALLEL		198.45	These are factory prime chips from same shipments we use in our professional quality boards.		
CABLE SETS		18.95	EXPORT NOTES		
VIDEO BOARDS			FOR 50Hz 230V C. V. POWER SUPPLY		Add 30.00
64 or 32 x 16		\$198.71	80 x 24 VIDEO BOARDS — Specify Format (No Added Charge)		
80 x 24 WITHOUT PROGRAMMABLE CHARACTER SET RAM		398.74	On orders under \$250.00 for a Single Board, or Chips, please Add \$30.00 Handling and we will ship Air Mail Prepaid.		
80 x 24 ALL FEATURES		458.76	On all other orders we will ship via Emery Air Freight Collect, and we will charge no handling.		
			All orders must be prepaid in U. S. Funds. Please note that foreign checks have been taking about eight weeks for collection, so we would advise wiring money or checks drawn on a bank account in the U. S. Our bank is the Continental Illinois National Bank of Chicago, account #73-32033. Visa or Master Charge also accepted.		

NOTE: OUR PRICE IS OUR CATALOG NUMBER. PLEASE BE SURE TO PRICE OUT COMPLETELY.

SMOKE SIGNAL BROADCASTING

Presents

3 Powerful New SS-50/SS-50C Boards

DCB-4 **Disk Master** Double Density Controller Board and DOS68D Double Density DOS **\$449.00**

The new DCB-4 is a truly state-of-the-art development which allows up to 366K bytes to be stored on a single 5¼" disk and has these outstanding features:

- Up to four 5¼" and four 8" drives can be handled in the same system with a user definable logical unit table. (DOS68D will be compatible with future hard disk system).
- Under software control, the user can select the following for any drive:
 - ☆ Single sided or double sided operation.
 - ☆ Single density or double density data.
 - ☆ 5¼" or 8".
 - ☆ Stepping Rate.
 - ☆ 40 track or 35 track density on double sided 5¼" drives.
 - ☆ User can select the system boot configuration.
- Occupies only 16 bytes of memory space (F760-F76F standard). User selectable to any 16 byte address space.
- Can read and write a single sector by itself. On-board buffer memory allows full interrupt capability in interrupt driven systems. Once data transfer has been initiated, no more processor time is required.
- Contains extended decoding circuitry for extended addressing per SS-50C bus which can be enabled by an option jumper.
- SSB provides a means for copying software written by older versions of DOS68 to be read by DOS68D. All new media formatted by DOS68D can be read by all older versions of DOS68. DOS68 is SSB's 6800 disk operating system.
- Track 0 of side 0 is recorded in single density per IBM standard.
- Phase-locked-loop assures highest data integrity attainable.

All of these features are available for immediate delivery on one standard 5¼" x 9" 50 pin SS-50/SS-50C card for only \$449.00. The price includes DOS68D version 5.1, MONITOR object code on diskette, and a manual with the source listing.

SCB-69 **Super Computer Board** 6809CPU Board **\$299.00**

The most versatile 6809 CPU Board on the market is now available from Smoke Signal Broadcasting and has the following features:

- Standard 2 MHz operation.
- 20 bit address generation for up to 1 Mbyte of memory. Uses an improved address translation RAM which is compatible with present extended addressing schemes yet requires much less overhead when used in multi-user systems.
- All on-board devices can be switch selected to occupy any or all extended pages. Any on-board device may be disabled and its memory space is then available for external memory.
- Standard real-time clock (time-of-day, day-of-week, day-of-month) with battery back up capable of generating programmable interrupts.
- Up to 20K of EPROM can be installed on the CPU Board.
- Standard 1K of RAM on board.
- Includes improved 6809 Monitor (and source listing).
- Contains an FPLA for decoding EPROM address and optional devices. Switches are used to select 2K/4K EPROM and Fast/Slow I/O.
- Contains provision for optional 9511/9512 floating point processor.
- NMI line is user selectable to work with either SS-50 or SS-50C busses.

Price for the new SCB-69 is only \$299.00 for an assembled, burned-in fully tested board.

M-32-X **32K** **Memory Board** **\$539.00 \$439.00**

The first and only 32K Static Ram Board on standard size (5½" x 9") SS-50/SS-50C Bus Circuit Card is made by Smoke Signal.

- Switch selectable to any 4K boundary.
- Any 4K block may be switch enabled or disabled.
- Fully compatible with SS-50C extended addressing (allows memory decoding up to 1 Mbyte).
- Extended addressing capability may be switched off for compatibility with SS-50 systems.
- Gold Bus Connectors for high reliability.
- Guaranteed 2MHz operation (tested at 2.2 MHz).
- Low power consumption — 8 volts at 2.4 amps typical.

M-32-X 32K Memory Board is priced at \$539.00.

M-24-X 24K Memory Board expandable to 32K, is \$439.00.

And our M-16-X 16K board is back to the old price of \$299.00.

SMOKE SIGNAL



BROADCASTING®

31336 Via Colinas, Westlake Village, CA 91361, (213) 889-9340

In the world of 6800 Microcomputing there is only one Universal Mini-Disk System ...

the PERCOM LFD-400™
with SOFTRAN™

Made possible by SOFTRAN™, an innovative \$24.95 translator program, the reliable Percom LFD-400™ has just been upgraded to the first universal mini-disk storage system.

Suddenly the two worlds of 6800 minidiskette software become one. Because the LFD-400™ with SOFTRAN™ can read either soft-sectored or hard-sectored disks.

And owning an LFD-400/SOFTRAN system means you can run minidiskette programs from the enormous combined selection of all of the principal 6800 software houses — TSC, Computerware, the Software Works, Hemenway Associates and of course Percom.

Available in versions for mini FLEX[†], FLEX 2.0[†] and Smoke Signal Broadcasting Company's DOS, SOFTRAN™ copies soft-sectored minidiskettes track-for-track onto hard-sectored minidiskettes. If the source disk includes a FLEX[†] or 'Smoke' DOS, SOFTRAN™ is used to modify the operating system to function with the Percom LFD-400™.

SOFTRAN™ is supplied on a minidiskette along with utilities for only \$24.95. A users manual is included. You must indicate whether SOFTRAN™ is to be used for mini FLEX[†], FLEX 2.0[†] or Smoke's DOS.

The Percom LFD-400™ mini-disk system sells for

only \$599.95, complete with: (1) the drive, drive electronics and Percom's rugged PS-401 power supply all in a finished enclosure, (2) a demonstrably superior controller PC card featuring an explicit data/ clock separation circuit, MPX, a remarkable 2K DOS, and provision for 1K extra PROM, (3) an interconnecting cable and (4) a 70-page users manual.

Also available: Upgrade kits for SWTP or 'Smoke' mini-disk drive systems. Kit includes LFD-400™ controller, MPX DOS & SOFTRAN™. Only \$224.95.

Available soon!

SOFTRAN™ for Percom's 77-track LFD-800™ mini-disk system; SOFTRAN/9™ for 6809 FLEX[†] files and programs.

[†] trademark of Technical Systems Consultants, Inc.



™ trademark of Percom Data Company, Inc.

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

Percom 'peripherals for personal computing'

To place an order or request additional literature call toll-free 1-800-527-1592. For technical information call (214) 272-3421. Orders may be paid by check, money order, CDD or charged to a VISA or Master Charge account. Texas residents must add 5% sales tax.

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE

A Few Extraordinary Products for Your 6800/6809 Computer

SS-50 Bus LFD-400™ and LFD-800™ Systems



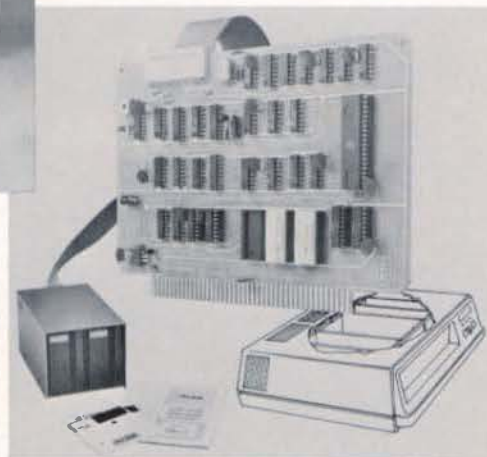
Percom mini-disk systems start as low as \$599.95, ready to plug in and run. You can't get better quality or a broader selection of disk software from any other microcomputer disk system manufacturer — at any price!

Features: 1-, 2- and 3-drive systems in 40- and 77-track versions store 102K- to 591K-bytes of random access data on-line • controllers include explicit clock/data separation circuit, motor inactivity time-out circuit,

buffered control lines and other mature design concepts • ROM DOS included with SS-50 bus version — optional DOSs for EXORciser® bus • extra PROM sockets on-board • EXORciser® bus version has 1K-byte RAM • supported by extended disk operating systems; assemblers and other program development/debugging aids: BASIC, FORTRAN, Pascal and SPL/M languages; and, business application programs.

From Percom . . .

**Low Cost
Mini-Disk Storage
in the Size You Want**



EXORciser® Bus LFD-400EX™ -800EX™ Systems



Upgrade to 6809 Computing Power

This 6809 upgrade adapter may be used on the SWTP 6800 and most other 6800/6802 MPU cards. Supplied assembled and tested, it costs only \$69.95 with user instructions. The original system may be restored by merely unplugging the adapter and a wire-jumpered DIP header, and re-inserting the original components. Also available for your upgrade computer is PSYMON™, the Percom SYstem MONitor for the Percom 6809 single-board computer. PSYMON™ on 2716 ROM costs only \$69.95 — PSYMON™ is also available on minidiskette, with source and object files, from the Percom Users Group.

™ trademark of Percom Data Company, Inc.

* trademark of the Motorola Corporation

Prices and specifications subject to change without notice.

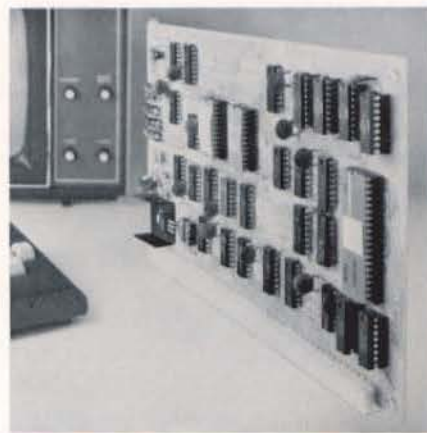
Full Feature Prototyping PC Boards

Percom SS-50 and I/O bus prototyping cards include all of the features needed for easy, straightforward prototyping. Use wire wrap, wiring pencil or solder wiring. **Features:** tin-lead plating over 2 oz. copper wets quickly, solders easily • provision for power regulators and distributed capacitor bypassing • SS-50 bus card accommodates 34- and

50-pin ribbon connectors on top edge, 10-pin Molex connector on side edge — costs only \$24.95 • I/O bus card is 1-1/4" higher than SWTP I/O card, accommodates 34-pin ribbon connector and 12-pin Molex connector on top edge — costs only \$14.95 • Both card designs accept 14-, 16-, 24- and 40-pin DIP sockets.

The Electric Window™: Instant, Real-Time Video Display Control

This VDC card resides completely in main memory so that control is accomplished instantly by direct MPU access to the on-board 2K character-store memory and the display control registers. Price is only \$249.95. **Features:** Programmable CRT controller chip provides extraordinary capability for software control of functions such as number of characters per line, number of lines displayed, highlighting and interlaced or non-interlaced scan • includes ASCII 128-unit character generator which generates 7-dot by 12-dot characters — lower case letters have descenders • provision for optional ROM for special characters/symbols • comprehensive manual includes full listing of WINDEX™, the Electric Window™ driver program — WINDEX™ is also available on minidiskette through the Percom Users Group.



PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

Products are available at Percom dealers nationwide. Call toll-free, 1-800-527-1592, for the address of your nearest dealer, or to order direct.

Multi-user Educational System on the /09

George Gerhold and Larry Kherlaty
Micropl, 2445 N. Nugent, Lummi Is. Wa, 98262

At Micropl we have developed a multi-user systems software package which runs on the SwTPC /09 (56K) computer. Although this system was designed for the classroom, the features make the system useful wherever a number of relatively unsophisticated computer users or a number of relatively simple applications must be served simultaneously.

A classroom computer system should provide three services: an author language of the highest quality for Computer-Assisted Instruction (CAI), a standard computational language for use in data analysis and in the teaching of programming, and a text editor for use in program preparation and in the teaching of word processing. High quality CAI is very rich in user interaction, as such it makes rather modest demands on the CPU as much time is spent waiting for user responses. High quality CAI has an extremely complex branching structure which requires a disk. The hardware requirements for the other two applications are essentially the same as those for CAI as students learning to program spend most of their time entering and correcting programs and comparatively little time actually running them. The combination of modest CPU demand with the need for a disk suggests a multi-user system. The Micropl Educational System is such a system.

The author language used for CAI applications is COMMON PILOT. COMMON PILOT is the most widely implemented CAI language; previous Micropl implementations include the 6800, North Star, CP/M, Helios, TRS-80, TERAk, TERC, Alpha, Interdata, and PASCAL versions. The 6809 version is the most advanced version; it includes instructions for graphics, file manipulation, and a full range of numeric (floating point and scientific) and string operations and functions. Because many readers are not familiar with the concept of author languages we will give a brief account of their characteristics. An author language is designed to make it easy for an author who is not primarily a programmer to program the computer to engage a user (student) in a dialogue. A crucial feature of any author language is the ability to recognize keywords imbedded in user responses. In a good author language it is very easy to allow for case and space editing, for spelling errors, for multiple answers, and for alternate answers - in all combinations. COMMON PILOT handles all this with a single instruction. Other necessary features include graphic display, easy positioning of text, execution of instructions generated at run time, full computational power, record keeping, continuation of program execution after program errors, and non-specification of input type (numeric or string). Also micro versions of author languages should execute programs directly from the disk so that the size of the memory will impose no limit on the program length.

COMMON PILOT includes all of the features listed above. A complete description of COMMON PILOT would require several articles the length of this one, so we will restrict ourselves here to a brief outline of the language. The fundamental instruction type is specified by a one or two letter op code. The op code may be modified by one or more single letter modifiers, and its execution may be made conditional by attaching a BASIC syntax expression to the op code. The colon is used to separate the op code (and modifiers or conditional) from the text field. For example,

T(C=1):Everything after the colon will be typed if C=1. The op codes are as follows:

OP CODE	FUNCTION	TEXT FIELD
T:	type	message to be typed
:	continue	continuation of message to be typed
A:	accept	optional, variables to be set from response
M:	match	keywords to compare with response
J:	jump	destination
U:	use	subroutine to be used
E:	end	optional, return point after subroutine
C:	compute	assignment code or edit string codes
R:	remark	remark
D:	dimension	string length or array size
G:	graphic	vector or point coordinates
XI:	execute	string where instruction stored
FI:	file in	record number and destination
FO:	file out	record number and source
PR:	problem	auto options on responses etc.

The modifiers are:

H	suppress line feed with type
J	automatic jump with match
S	spelling correction on match
X	suppress editing on response
Y	execute only if last match successful
N	execute only if last match unsuccessful
Digit	execute if same value as answer counter
E	execute if error flag set
C	execute if last relational expression true

The special characters which can be used in the text field of M: are

*	match any single character
&	match any number of unspecified characters
%	match space or either end of response
	or operator (& is and operator)

The computational features are:

Operators	+ - * / ** (concat)
Relational	= < > <= >= &(and) ~(not) l(or)
Functions	ABS FIX INT SGN RND SIN COS ATN SQR EXP LOG LN STR FLO ASC CHR LEN INS

The subscripting allows one and two dimensional numeric arrays of maximum dimension 999 and strings of maximum length 999. Substrings are specified by the notation A\$(position,length), and substrings can be used on either side of the assignment operator =.

This is not a complete list of the features of COMMON PILOT, but it should be sufficient to justify our claim that COMMON PILOT is the most powerful author language available for microcomputers and our claim that it is competitive with author languages used on the largest computers.

Because the expressions in COMMON PILOT are written in BASIC syntax we were able to create a version of BASIC based on this expression processor. We call this version Student BASIC because it was designed for use in teaching beginning students the fundamentals of programming. Variables, arrays, strings, dimensioning, loops, conditionals, and subroutines are all included. The following is a summary of the features found in Student BASIC. The elements of the expressions are the same as those listed earlier in the summary of COMMON PILOT. The statement types in the current version are

DIM	INPUT
END	LET
FOR ... NEXT	REM
GOSUB	RETURN
GOTO	STOP
IF ... THEN	

Line numbers are not required; they serve only as destinations for GOTO and GOSUB.

A number of extensions to the Student BASIC are planned. The most notable of these is file manipulation. We want to make it clear that we do not intend to expand Student BASIC into a full-feature BASIC. This will remain a teaching version which can also be used for multiple terminal input.

The text editor is a screen-oriented editor based on the features of the SWTPC CT-82 terminal. The screen functions as a 16 line window which can be shifted forward or backward through the file. All editing is done on the bottom line of the screen. Overtyping replaces characters; insert and delete have the obvious effect. The following is a summary of the editor functions.

Scroll up, Up arrow, Return	- move text up one line
Scroll down, Down arrow	- move text down one line
Right arrow, Left arrow	- move cursor
Backspace	- move cursor left and erase
Insert	- insert space, bump text right
Delete	- remove character, bump left
Form	- change case mode
Xmit	- leave editor, prompt for save

Again some extensions are planned. It will be possible to list the disk directory from any terminal, and it will be possible to direct output to a printer.

This text editor can be used to create or edit text files, that is FLEX files with the .TXT extension. Both the COMMON PILOT and the Student BASIC interpreters execute such files as programs. If a computation is too complex or too lengthy for the Student BASIC, programs can be entered using this text editor in the multi-user mode, and subsequently the programs can be run in single-user mode using any of the TSC 6809 BASICS. The compiler versions are probably the best for this purpose as neither the editor nor those compilers require line numbers.

When the system is initialized a prompt line is displayed at each terminal. This gives each user the choice of one of the following activities:

- Run a COMMON PILOT program
- Run a Student BASIC program
- List a text file
- Create a new text file
- Edit an existing text file

In addition the master terminal can return the whole system to FLEX.

The hardware requirements for the system are

- 6809 Computer
- Dual Disk Drives (8" preferred for multiuser)
- 1 to 4 CT-82 Terminals
- 1 to 4 Serial Interfaces
- MP-N Numeric Processor
- 32K (1 user) to 56K (4 users)
- FLEX DOS

The system is supplied in a four terminal configuration with the assumed memory size of 56K. This provides an 8K workspace for each terminal. The workspace is used mainly for array, label, and string

storage in COMMON PILOT and Student BASIC and for text storage in the text editor. No alterations are needed with fewer terminals. Instructions are included for reallocation of the workspace as there is no reason that the partitions need be of the same size.

The Micropl Educational System can be purchased direct from Micropl, from SWTPC, and from selected dealers. The system is provided with an 85 page COMMON PILOT manual, with a 14 page manual on Student BASIC and the text editor, and with a variety of demonstration CAI programs. The software can be supplied on FLEX compatible disks (5" or 8"). The cost of the package is \$250. When they become available the extensions mentioned will be provided at no additional cost to purchasers who return a Micropl Educational System disk to us.

William Stock -----
1125 Lois Dr.
Cincinnati, OH
45237

BOOKKEEPING=Part 2 MINIflex™
Continued from last month.

Note: The source listing are of such length that we will run additional listing next month. We dedicate this space due to the large desire, on the part of many readers, for this type software. For those running FLEX™ 2.0 the conversion should be simple. For the small business it can be enlarged to fit most needs for a simple system.

```
0001 REM START.BAS
0002 REM THIS PROGRAM GETS & VALIDATES
0003 REM TODAY'S DATE, STORES IT IN
0004 REM THE PARAMETERS FILE,
0005 REM AND CALLS IN THE MENU
0006 REM PROGRAM
0007 REM
0010 OPEN 01,0.PRM.DAT
0020 READ 01,V1,V2,V3,V4,V5,V6,V7,V8,V9
0025 CLOSE 01
0030 PRINT CHR$(16);CHR$(22);CHR$(0);CHR$(0);CHR$(0);
0040 INPUT "TODAY'S DATE (MM/DD/YY)",V1
0043 REM
0045 REM VALIDATE INPUT
0047 REM
0050 A=INT(V1/10000)
0053 IF A<1 THEN 40
0056 IF A>12 THEN 40
0060 B=INT(V1/100)-A*100
0063 IF B<1 THEN 40
0066 IF B>31 THEN 40
0067 REM
0068 REM CALCULATE DAY
0069 REM
0070 C=V1-B*100-A*10000
0080 ON A GOSUB 200,210,220,230,240,250,260,270,280,290,300,310
0090 IF INT(C/4)=C/4 THEN IF A>2 THEN IF C>0 THEN B=B+1
0100 B=B+C*INT((C-1)/4)/7
0110 B=INT((B-INT(B)/7)*7+1,5)
0120 ON B GOSUB 1200,1210,1220,1230,1240,1250,1260
0123 REM
0125 REM RIGHT DAY?
0127 REM
0130 INPUT "TODAY'S DAY (MON, TUE)",B0
0150 IF A6<>LEFT$ (B0,3) THEN 30
0160 OPEN 01,0.PRM.DAT
0165 SCRATCH 01
0170 WRITE 01,V1,V2,V3,V4,V5,V6,V7,V8,V9
0180 CLOSE 01
0190 CHAIN 0.MENU.BAS
0200 RETURN
0210 B=B+31:RETURN
0220 B=B+59:RETURN
0230 B=B+90:RETURN
0240 B=B+120:RETURN
0250 B=B+151:RETURN
0260 B=B+181:RETURN
0270 B=B+212:RETURN
0280 B=B+243:RETURN
0290 B=B+273:RETURN
0300 B=B+304:RETURN
0310 B=B+334:RETURN
1200 A$="SUM":RETURN
1210 A$="MON":RETURN
1220 A$="TUE":RETURN
1230 A$="WED":RETURN
1240 A$="THU":RETURN
1250 A$="FRI":RETURN
1260 A$="SAT":RETURN
0001 REM THIS PROGRAM LOADS THE SYSTEM
0002 TO RETURN.
0003 REM MENU.BAS
0004 REM THE PARAMETERS FILE,
0005 REM AND CALLS IN THE MENU
0006 REM PROGRAM
0007 REM
0010 OPEN 01,0.PRM.DAT
0020 READ 01,V1,V2,V3,V4,V5,V6,V7,V8,V9
0025 CLOSE 01
```

```
0030 PRINT CHR$(16);CHR$(22);CHR$(0);CHR$(0);CHR$(0);
0040 INPUT "TODAY'S DATE (MM/DD/YY)",V1
0043 REM
0045 REM VALIDATE INPUT
0047 REM
0050 A=INT(V1/10000)
0053 IF A<1 THEN 40
0056 IF A>12 THEN 40
0060 B=INT(V1/100)-A*100
0063 IF B<1 THEN 40
0066 IF B>31 THEN 40
0067 REM
0068 REM CALCULATE DAY
0069 REM
0070 C=V1-B*100-A*10000
0080 ON A GOSUB 200,210,220,230,240,250,260,270,280,290,300,310
0090 IF INT(C/4)=C/4 THEN IF A>2 THEN IF C>0 THEN B=B+1
0100 B=B+C*INT((C-1)/4)/7
0110 B=INT((B-INT(B)/7)*7+1,5)
0120 ON B GOSUB 1200,1210,1220,1230,1240,1250,1260
0123 REM
0125 REM RIGHT DAY?
0127 REM
0130 INPUT "TODAY'S DAY (MON, TUE)",B0
0150 IF A6<>LEFT$ (B0,3) THEN 30
0160 OPEN 01,0.PRM.DAT
0165 SCRATCH 01
0170 WRITE 01,V1,V2,V3,V4,V5,V6,V7,V8,V9
0180 CLOSE 01
0190 CHAIN 0.MENU.BAS
0200 RETURN
0210 B=B+31:RETURN
0220 B=B+59:RETURN
0230 B=B+90:RETURN
0240 B=B+120:RETURN
0250 B=B+151:RETURN
0260 B=B+181:RETURN
0270 B=B+212:RETURN
0280 B=B+243:RETURN
0290 B=B+273:RETURN
0300 B=B+304:RETURN
0310 B=B+334:RETURN
1200 A$="SUM":RETURN
1210 A$="MON":RETURN
1220 A$="TUE":RETURN
1230 A$="WED":RETURN
1240 A$="THU":RETURN
1250 A$="FRI":RETURN
1260 A$="SAT":RETURN
```

```
0010 DIM X(2,10):A$="APIV.BAS"
0020 GOSUB 1000
0030 PRINT CHR$(16);CHR$(22);CHR$(0);CHR$(0);
0040 PRINT "SEQ $ -I:V2;I:0:15";"POSTING"
0050 PRINT
0060 PRINT "ACCT AMOUNT";TAB(17);"DESCRIPTION"
0070 GOSUB 1150:REM INPUT
0080 IF A$="" THEN 400
0090 IF A$=V8 THEN 110
0100 PRINT "ACCESS NOT ALLOWED":GOTO 330
0110 IF A$=V6 THEN B=-B
0120 B(1)=A$;B(1,1)=A$;B(2,1)=B*ABS(B)
0130 FOR X=2 TO 10
0140 B(2,X)=0
0150 NEXT X
0160 IF LEN(A$)<14 THEN PRINT
0170 FOR X=2 TO 10
0180 GOSUB 1150:REM INPUT
0190 IF A$="" THEN X=10:GOTO 290
0200 IF A$=V8 THEN PRINT "ACCESS NOT ALLOWED":X=10:GOTO 290
0203 REM
0205 REM DETERMINE DEBIT/CREDIT
0207 REM
0210 IF B(1,1)<=V4 THEN GOSUB 500:GOTO 240
0220 IF B(1,1)<=V5 THEN GOSUB 540:GOTO 240
0230 IF B(1,1)<=V6 THEN GOSUB 570:GOTO 240
0240 IF B(1,1)<=V7 THEN GOSUB 610:GOTO 240
0250 GOSUB 650
0260 B(1,X)=A$;B(2,X)=B(1,X)*A$
0270 P=P-ABS(B)
0280 IF P=0 THEN X=10:REM IN BALANCE
0285 IF B=0 THEN X=10
0290 NEXT X
```

```

0300 GOSUB 1310:REM PRINT TRANSACTION
0310 IF P=0 THEN INPUT "EVERYTHING OK (Y/N)",A$:GOTO 340
0320 PRINT "TRANSACTIONS NOT IN BALANCE"
0330 INPUT "PRESS RETURN",A$:GOTO 30
0340 IF LEFT$(A$,1)="Y" THEN 370
0350 IF LEFT$(A$,1)="N" THEN 30
0360 GOTO 310
0370 GOSUB 2000:REM WRITE TRANS
0380 V=V2+1
0390 GOTO 30
0400 INPUT "IS JOB TERMINATED",A$
0410 IF LEFT$(A$,1)(">"") THEN 30
0412 REM
0413 REM CLOSE FILED, UPDATE PARAMETERS, CHAIN NEXT PROGRAM
0414 REM
0420 CLOSE #2 :IF F=0 THEN WRITE #1,F
0425 CLOSE #1
0430 OPEN #1,O,PRN,BAT
0440 SCRATCH #1
0450 WRITE #1,V1,U2,V3,V4,V5,V6,V7,V8,V9
0460 CLOSE #1
0470 IF F=1 THEN CHAIN O.PRAT
0480 CHAIN JAPP
0490 REM
0491 REM DEBIT/CREDIT SUBROUTINES
0492 REM
0500 IF A<V4 THEN 680
0510 IF A<V6 THEN 690
0520 IF A<V7 THEN 680
0530 GOTO 690
0540 IF A<V4 THEN 690
0545 IF A<V6 THEN 680
0550 IF A<V7 THEN 690
0560 GOTO 680
0570 IF A<V4 THEN 690
0580 IF A<V6 THEN 680
0590 IF A<V7 THEN 690
0600 GOTO 680
0610 IF A<V4 THEN 680
0620 IF A<V6 THEN 690
0630 IF A<V7 THEN 680
0640 GOTO 690
0650 IF A<V4 THEN 690
0660 IF A<V6 THEN 680
0670 IF A<V7 THEN 690
0680 B=BOH(B(2,1)):ABO(B(2)):RETURN
0690 B=BOH(R(2,T)):ABO(B(2)):RETURN
1000 PRINT CHR$(16);CHR$(22);CHR$(0);CHR$(0);CHR$(0);
1010 INPUT "IS LATEST JOURNAL ON B1",A$
1020 IF LEFT$(A$,1)(">"") THEN PRINT "PUT IT ON":GOTO 1010
1030 OPEN #1,O,PRN,BAT
1040 REMB #1,V1,U2,V3,V4,V5,V6,V7,V8,V9
1050 CLOSE #1
1060 KILL 1..TRAN.BAT
1070 KILL O.PTRAN.BAT
1100 OPEN #1,O,PTRAN
1110 OPEN #2,1..TRAN
1120 RETURN
1140 REM
1143 REM INPUT ROUTINE
1144 REM
1150 S(1)=0:G(2)=0:INPUT A$
1160 IF A$="" THEN RETURN
1170 IF ABC(A$)>57 THEN A$="":RETURN
1180 A=VAL(A$):B=0
1190 Z=1:FOR T=1 TO LEN(A$)
1200 IF MID$(A$,T,1)="" THEN B(Z)=T:Z=Z+1
1210 NEXT T
1220 A$=RIGHT$(A$,LEN(A$)-B(1))
1230 IF ABC(A$)>57 THEN A$="":RETURN
1240 B=VAL(A$):A=ABO(A)
1250 IF S(2)=0 THEN A$=CHR$(19):RETURN
1260 A$=R(HTO(A$,LEN(A$)):B(1)-B(2))
1270 RETURN
1299 REM
1300 REM PRINT SCREEN
1301 REM
1310 PRINT CHR$(16);CHR$(22);CHR$(0);CHR$(0);CHR$(0);
1320 PRINT "ACCT CREDIT DEBIT"
1330 X=1:GOSUB 1500
1340 GOSUB 1600
1350 PRINT
1360 FOR X=2 TO 10
1370 IF D(2,X)=0 THEN X=10:GOTO 1400
1380 GOSUB 1500
1390 GOSUB 1600
1400 NEXT X
1410 RETURN
1490 REM
1493 REM BEAUTIFY $ AMOUNTS
1495 REM
1500 A$=STR$(ABO(B(2,X)))
1510 IF D(2,X)=INT(D(2,X)) THEN A$=A$+"."
1520 IF D(2,X)=10:INT(D(2,X))=10 THEN A$=A$+"0"
1530 RETURN
1600 PRINT D(1,X);TAB(15-LEN(A$));
1610 IF D(2,X)<0 THEN PRINT TAB(POB+10);
1620 PRINT A$
1630 RETURN
2000 FOR X=1 TO 10

```

```

2010 IF D(2,X)=0 THEN X=10:GOTO 2030
2013 REM
2015 REM TRAN FILE
2017 REM
2020 WRITE #2,D(1,X),V2,V1,D(2,X),D(3,X)
2030 NEXT X
2040 FOR X=1 TO 10
2050 IF D(2,X)=0 THEN X=10:GOTO 2080
2053 REM
2055 REM A/P TRAN FILE
2057 REM
2060 IF D(1,X)>V7 THEN WRITE #1,D(1,X),D(2,X),V2
2070 IF D(1,X)>V7 THEN F=1
2080 NEXT X
2090 RETURN

0010 GOTO 3000
0090 REM PSAT.BAS
0093 REM SUBROUTINED FIRST FOR
0095 REM FASTER ACCESS
0097 REM
0098 REM SHEL-NESTING GORS
0099 REM
0100 I=L
0110 J=INT(I/2)
0120 IF J=0 THEN RETURN
0130 I=J
0140 T=I
0150 B=T+1
0160 IF D(1,T)<B(1,B1) THEN 190
0170 GOSUB 2101 T=T-I
0180 IF T>0 THEN 150
0190 I:=I+T:IF B(1,T)>I THEN 110
0200 GOTO 140
0210 C=B(1,T):D(1,T)=B(1,B1):B(1,T)=C
0220 C=B(2,T):D(2,T)=B(2,B1):B(2,T)=C
0240 RETURN
0243 REM
0245 REM VARIOUS FILE-HANDLING SUBROUTINES
0247 REM
0250 OPEN #2,SU1:RETURN
0260 OPEN #2,SU2:RETURN
0270 OPEN #2,SU3:RETURN
0280 OPEN #2,SU4:RETURN
0290 OPEN #2,SU5:RETURN
0300 OPEN #2,SU6:RETURN
0310 OPEN #2,SU7:RETURN
0320 OPEN #2,SU8:RETURN
0330 KILL SUB.BAT
0340 KILL SU7.BAT
0350 KILL SU4.BAT
0360 KILL SU3.BAT
0370 KILL SU4.BAT
0380 KILL SU3.BAT
0390 KILL SU2.BAT
0400 KILL SU1.BAT:RETURN
0410 OPEN #9,SU8
0420 OPEN #8,SU7
0430 OPEN #7,SU6
0440 OPEN #6,SU5
0450 OPEN #5,SU4
0460 OPEN #4,SU3
0470 OPEN #3,SU2
0480 OPEN #2,SU1:RETURN
0510 REMB #2,D(1,C),D(2,C),D(3,C):IF EOF(2)=1 THEN 590
0515 RETURN
0520 REMB #3,D(1,C),D(2,C),D(3,C):IF EOF(3)=1 THEN 590
0525 RETURN
0530 REMB #4,D(1,C),D(2,C),D(3,C):IF EOF(4)=1 THEN 590
0535 RETURN
0540 REMB #5,D(1,C),D(2,C),D(3,C):IF EOF(5)=1 THEN 590
0545 RETURN
0550 REMB #6,D(1,C),D(2,C),D(3,C):IF EOF(6)=1 THEN 590
0555 RETURN
0560 REMB #7,D(1,C),D(2,C),D(3,C):IF EOF(7)=1 THEN 590
0565 RETURN
0570 REMB #8,D(1,C),D(2,C),D(3,C):IF EOF(8)=1 THEN 590
0575 RETURN
0580 REMB #9,D(1,C),D(2,C),D(3,C):IF EOF(9)=1 THEN 590
0585 RETURN
0590 B(1,C)=99999999:RETURN
0610 CLOSE #9
0620 CLOSE #8
0630 CLOSE #7
0640 CLOSE #6
0650 CLOSE #5
0660 CLOSE #4
0670 CLOSE #3
0680 CLOSE #2:RETURN
0690 REM
0693 REM SET B ITEMS
0695 REM
0700 READ #1,N
0710 IF EOF(1)=1 THEN 740
0720 F=F+1
0730 GOTO 700
0740 RESTORE #1
0750 IF F>240 THEN F=INT(F/8)+1:RETURN
0760 F=30
0770 RETURN

```

```

3000 OPEN B1,O.PTRAM.BAT 100000 3400
3010 DIR B(3,255):00000 700:REM DET B FILED
3020 M=0
3030 FOR X=1 TO F
3040 READ B(1,X),B(2,X),B(3,X)
3050 L=X
3060 IF EOF(1)=1 THEN L=L-1:Z=F
3070 NEXT X
3080 IF L=0 THEN J170
3090 GOSUB 100:REM SORT THIS BATCH
3100 M=M+1
3103 REM
3105 REM OPEN APPROPRIATE WORK FILE
3107 REM
3110 ON A GOSUB 250,260,270,280,290,300,310,320
3120 FOR X=1 TO L
3130 WRITE B2,B(1,X),B(2,X),B(3,X)
3140 NEXT X
3150 CLOSE B2
3160 IF EOF(1)=0 THEN 3030
3163 REM
3165 REM OPEN INPUT FILES FOR MERGE
3167 REM
3170 ON A GOSUB 400,470,460,450,440,430,420,410
3175 CLOSE A1
3180 OPEN B1,I.APST.BAT
3190 SCRATCH B1
3193 REM
3195 REM FILL TABLE FIRST TIME
3197 REM
3200 FOR C=1 TO M
3210 ON C GOSUB 510,520,530,540,550,560,570,580
3220 NEXT C
3223 REM
3225 REM MERGE SORTED INPUT FILES
3227 REM
3230 C=1:IF M=1 THEN 3270
3240 FOR X=2 TO M
3250 IF D(1,C)>D(1,X) THEN C=X
3260 NEXT X
3270 IF C(1,C)=99999999 THEN 3320
3280 WRITE B1,B(1,C),B(2,C),B(3,C)
3290 ON C GOSUB 510,520,530,540,550,560,570,580
3310 GOTO 3230
3320 CLOSE B1
3323 REM
3325 REM CLOSE & DELETE WORK FILES
3327 REM
3330 ON M GOSUB 600,670,660,650,640,630,620,610
3340 ON M GOSUB 400,390,380,370,360,350,340,330
3345 PRINT "PRINT (M-1)*F+L: "ITEM#
3350 INPUT "IS OLDEST A/P RATED ON 80",A9
3355 IF A9="" THEN 3350
3360 IF LEFT$(A9,1)<"Y" THEN PRINT "PUT IT ON":GOTO 3350
3370 CHAIN PUB
3400 PRINT CHR$(16);CHR$(22);CHR$(0);CHR$(0);CHR$(0);
3410 INPUT "IS LATEST A/P RATED ON 81",A9
3415 IF A9="" THEN 3410
3420 IF LEFT$(A9,1)<"Y" THEN PRINT "PUT IT ON":GOTO 3410
3430 RETURN

```

```

0001 REM PAYABLE UPDATE P.M.D.BAS
0010 PRINT CHR$(14);CHR$(27);CHR$(0);CHR$(0);CHR$(0);
0020 PRINT "PAYABLES UPDATING"
0030 KILL O. PHASTED.BAT
0040 OPEN B2,APST
0050 OPEN B4,I.APMASTER
0060 OPEN B5,O.APMASTER
0070 READ B4,P1,P2,P3,P4
0080 READ B2,A,B,C
0090 IF E0(4)=1 THEN P1=99999999
0100 IF EOF(2)=1 THEN A=99999999
0200 IF A>P1 THEN 270
0210 IF A<P1 THEN 320:REM SEQUENCE ERROR/ACCT NOT ON FILE
0220 IF EOF(2)=1 THEN 270
0230 P2=P2+B
0240 READ B2,A,B,C
0250 IF EOF(2)=1 THEN A=99999999
0260 GOTO 200
0270 IF EOF(4)=1 THEN 400
0280 WRITE B5,P1,P2,P3,P4
0290 READ B4,P1,P2,P3,P4
0300 IF (B(4)=1) THEN P1=99999999
0310 GOTO 200
0320 PRINT "ERROR: ACT ";A;"-SER ";C
0330 E=E+1:GOTO 240
0400 CLOSE B2,B4,B5
0410 IF E>0 THEN 450:REM ABORT
0415 PRINT "LATEST A/P IS ON 80"
0420 PRINT "INPUT "IS LATEST JOURNAL ON 81",A9
0430 IF LEFT$(A9,1)<"Y" THEN PRINT "PUT IT ON":GOTO 420
0440 CHAIN JAPP
0450 PRINT E:"ERRORS."
0460 PRINT "RE-ENTER ALL DATA AFTER"
0470 PRINT "CORRECTION PROBLEM."
0480 PRINT "PRINT "PUT BYTES 018X ON 80"
0490 INPUT "IS IT OK",A9
0495 IF LEFT$(A9,1)<"Y" THEN 400
0500 CHAIN B.NEWM

```

```

0001 REM APPEND TRANSACTIONS TO JOURNAL
0002 REM JAPP.BAS
0010 PRINT CHR$(16);CHR$(22);CHR$(0);CHR$(0);CHR$(0);
0020 INPUT "IS OLDEST JOURNAL ON 80",A9
0030 IF LEFT$(A9,1)<"Y" THEN 20
0040 PRINT
0050 PRINT "DON'T BOTHER ME"
0060 PRINT "I'M BUSY"
0070 KILL O.JOURNAL.BAT
0080 OPEN B1, JOURNAL
0090 OPEN B2, O.JOURNAL
0100 READ B1,A,B,C,D,A9
0110 IF EOF(1)=1 THEN 140
0120 WRITE B2,A,B,C,D,A9
0130 GOTO 100
0140 CLOSE B1
0150 OPEN B1,TRAN
0160 READ B1,A,B,C,D,A9
0170 IF EOF(1)=1 THEN 200
0180 WRITE B2,A,B,C,D,A9
0190 GOTO 140
0200 CLOSE B1,B2
0210 PRINT CHR$(16);CHR$(22);CHR$(0);CHR$(0);CHR$(0);
0215 PRINT "LATEST JOURNAL IS IN 80."
0220 PRINT "HOW I NEED THE LATEST S/L IN 80"
0230 PRINT
0240 INPUT "IS THE LATEST S/L IN 80",A9
0250 IF LEFT$(A9,1)<"Y" THEN 210
0260 CHAIN O.08R1.DAB

```

```

0001 REM GENERAL LEDGER SORT
0002 REM G.LRT.BAS
0008 REM SUBROUTINES FIRST
0009 REM
0010 GOTO 3000
0090 REM
0095 REM SHELLE-METED SORT
0097 REM
0100 I=L
0110 I=INT(1/2)
0120 IF I=0 THEN RETURN
0130 I1=I
0140 I2=I
0150 S=I+1
0160 IF B(I,T)<B(I1,T) THEN 190
0170 GOSUB 210:T=T-1:IF T>0 THEN 150
0180 I1=I+1:IF I1>L-1 THEN 110
0200 GOTO 140
0210 C=B(I,T):B(I,T)=B(I1,T):B(I1,T)=C
0220 C=B(2,T):B(2,T)=B(2,I1):B(2,I1)=C
0230 C=B(4,T):B(4,T)=B(4,I1):B(4,I1)=C
0240 RETURN
0243 REM
0245 REM VARIOUS FILE SUBROUTINES
0247 REM
0250 OPEN B2,SU1:RETURN
0260 OPEN B2,SU2:RETURN
0270 OPEN B2,SU3:RETURN
0280 OPEN B2,SU4:RETURN
0290 OPEN B2,SU5:RETURN
0300 OPEN B2,SU6:RETURN
0310 OPEN B2,SU7:RETURN
0320 OPEN B2,SU8:RETURN
0330 KILL SU8.BAT
0340 KILL SU7.BAT
0350 KILL SU6.BAT
0360 KILL SU5.DAT
0370 KILL SU4.DAT
0380 KILL SU3.DAT
0390 KILL SU2.DAT
0400 KILL SU1.BAT:RETURN
0410 OPEN B9,SU8
0420 OPEN B9,SU7
0430 OPEN B7,SU6
0440 OPEN B6,SU5
0450 OPEN B5,SU4
0460 OPEN B4,SU3
0470 OPEN B3,SU2
0480 OPEN B2,SU1:RETURN
0510 READ B(1,C),B(2,C),B(3,C),B(4,C):IF EOF(2)=1 THEN 590
0515 RETURN
0520 READ B(1,C),B(2,C),B(3,C),B(4,C):IF EOF(3)=1 THEN 590
0525 RETURN
0530 READ B(1,C),B(2,C),B(3,C),B(4,C):IF EOF(4)=1 THEN 590
0535 RETURN
0540 READ B(1,C),B(2,C),B(3,C),B(4,C):IF EOF(5)=1 THEN 590
0545 RETURN
0550 READ B(1,C),B(2,C),B(3,C),B(4,C):IF EOF(6)=1 THEN 590
0555 RETURN
0560 READ B(1,C),B(2,C),B(3,C),B(4,C):IF EOF(7)=1 THEN 590
0565 RETURN
0570 READ B(1,C),B(2,C),B(3,C),B(4,C):IF EOF(8)=1 THEN 590
0575 RETURN
0580 READ B(1,C),B(2,C),B(3,C),B(4,C):IF EOF(9)=1 THEN 590
0585 RETURN
0590 B(1,C)=99999999:RETURN
0610 CLOSE B9
0620 CLOSE B8

```

```

0630 CLOSE #7
0640 CLOSE #6
0650 CLOSE #5
0660 CLOSE #4
0670 CLOSE #3
0680 CLOSE #2 :RETURN
0690 REM
0693 REM COUNT R ITEMS, COMPUTE # FILES
0695 REM
0700 READ B1,R
0710 IF EOF(1)=1 THEN 740
0720 F=F+1
0730 GOTO 700
0740 RESTORE #1
0750 IF F>240 THEN F=INT(F/8)+1:RETURN
0760 F=30
0770 RETURN
3000 OPEN R1,TRAN
3010 DIM B(4,255):GOSUB 700:REM SET # FILES
3020 N=0
3023 REM
3025 REM GET A SORT ITEMS
3027 REM
3030 FOR X=1 TO F
3040 READ B1,B11,X1,B12,X1,B13,X1,B14,X1
3050 L=X
3060 IF EOF(1)=1 THEN L=L-1:X=F
3070 NEXT X
3080 IF L=0 THEN 3170
3090 GOSUB 100:REM SORT
3100 N=N+1
3103 REM
3105 REM OPEN APPROPRIATE WORK FILE
3107 REM
3110 ON N GOSUB 250,260,270,280,290,300,310,320
3120 FOR X=1 TO L
3130 WRITE B2,B11,X1,B12,X1,B13,X1,B14,X1
3140 NEXT X
3150 CLOSE B2
3160 IF EOF(1)=0 THEN 3030
3163 REM
3165 REM OPEN MERGE INPUTS
3167 REM
3170 ON N GOSUB 480,470,460,450,440,430,420,410
3180 CLOSE B1:KILL O,OLTRAN,BAT
3190 OPEN #1,O,OLTRAN
3193 REM
3195 REM FILL TABLE 1ST TIME
3197 REM
3200 FOR C=1 TO N
3210 ON C GOSUB 510,520,530,540,550,560,570,580
3220 NEXT C
3223 REM
3225 REM MERGE
3227 REM
3230 C=1:IF N=1 THEN 3270
3240 FOR X=2 TO N
3250 IF B11,C>B11,X1 THEN C=X
3260 NEXT X
3270 IF B11,C=99999999 THEN 3320
3280 WRITE B1,B11,C,B12,C,B13,C,B14,C
3290 ON C GOSUB 510,520,530,540,550,560,570,580
3300 REM GO BACK FOR NEXT ITEM
3310 GOTO 3230
3313 REM
3315 REM OPEN OLDFST FILE, REFORMAT WORK FILES
3317 REM
3320 CLOSE R1
3330 ON N GOSUB 680,670,660,650,640,630,620,610
3340 ON N GOSUB 400,390,380,370,360,350,340,330
3345 PRINT :PRINT (M-1)*F+L:"ITEMS"
3350 CHAIN O,OLDF.BAS

```

```

0001 REM THIS PROGRAM UPDATES THE GENERAL LEDGER &
0002 REM THE O/L HISTORY.
0003 REM OLDF.BAS
0004 REM IT WORKS. PLEASE DON'T
0005 REM BEFORE CHANGING ANYTHING.
0006 REM
0010 INPUT "IS OLDEST O/L ON B1",A0
0020 IF LEFT$(A0,1)<>"Y" THEN 10
0030 PRINT CHR$(16);CHR$(22);CHR$(0);CHR$(0);CHR$(0);
0040 KILL 1,OLMASTER,BAT
0050 KILL 1,OLHIST,BAT
0060 OPEN #1,O,OLHIST
0070 OPEN #2,O,OLTRAN
0080 OPEN #3,OLHIST
0090 OPEN #4,O,OLMASTER
0100 OPEN #5,OLMASTER
0110 DIM A(4,2)
0120 READ B1,A(1,1),A(2,1),A(3,1),A(4,1)
0130 READ B2,A(1,2),A(2,2),A(3,2),A(4,2)
0140 READ B4,B1,B2,B0
0150 IF EOF(1)=1 THEN A(1,1)=99999999
0160 IF EOF(2)=1 THEN A(1,2)=99999999
0200 IF A(1,1)>B1 THEN 250
0205 IF EOF(1)=1 THEN 250
0210 WRITE B3,A(1,1),A(2,1),A(3,1),A(4,1)

```

```

0220 READ B1,A(1,1),A(2,1),A(3,1),A(4,1)
0230 IF EOF(1)=1 THEN A(1,1)=99999999
0240 GOTO 200
0250 IF A(1,2)>B1 THEN 320
0260 IF A(1,2)<B1 THEN PRINT "ERROR: ACT ";A(1,2);"SEQ ";A(2,2);
0263 IF A(1,2)<B1 THEN PRINT "4";A(4,2):GOTO 290
0265 IF EOF(2)=1 THEN 320
0270 B2=B2+A(4,2)
0280 WRITE B3,A(1,2),A(2,2),A(3,2),A(4,2)
0290 READ B2,A(1,2),A(2,2),A(3,2),A(4,2)
0300 IF EOF(2)=1 THEN A(1,2)=99999999
0310 GOTO 250
0320 IF EOF(4)=1 THEN 400
0330 WRITE B5,B1,B2,B0,B3
0340 READ B4,B1,B2,B0,B3
0350 IF EOF(4)=1 THEN B1=99999999
0360 GOTO 200
0400 CLOSE B1,B2,B3,B4,B5
0405 PRINT "CURRENT O/L ON B1":PRINT
0410 INPUT "IF N SOME. HIT RETURN. ",A0
0420 PRINT CHR$(16);CHR$(22);CHR$(0);CHR$(0);CHR$(0);
0430 INPUT "IS LATEST DISC ON B0",A0
0440 IF LEFT$(A0,1)<>"Y" THEN 430
0450 CHAIN O,NEWU

```

```

0001 REM THIS IS THREE PROGRAMS -
0002 REM END YEAR, END PERIOD, JOURNAL PRINT.
0003 REM JPRINT.BAS
0005 STOP :END
0010 F=1:GOTO 100:REM END YEAR
0020 F=2:GOTO 100:REM END PERIOD
0030 F=3:GOTO 100:REM JOURNAL PRINT
0100 ON F GOTO 1000,1000,3000
0120 PRINT CHR$(16);CHR$(22);CHR$(0);CHR$(0);CHR$(0);
0130 RETURN
0190 REM
0193 REM END YEAR & END PERIOD
0195 REM
1000 GOSUB 120
1010 PRINT "PUT OLDEST JOURNAL ON B0"
1020 INPUT "IS IT THERE",A0
1030 IF A0="" THEN 1020
1040 IF LEFT$(A0,1)<>"Y" THEN 1010
1050 PRINT :INPUT "DO YOU WANT JOURNAL ON TAPE",A0
1060 IF A0="" THEN 1030
1070 IF LEFT$(A0,1)= "Y" THEN T=1:GOTO 1100
1080 IF LEFT$(A0,1)<>"N" THEN 1050
1090 T=2
1100 OPEN B1,O,JOURNAL.DAT
1120 SCRATCH B1
1130 WRITE B1,A
1140 CLOSE B1
1150 ON T GOTO 1160,1250
1160 POKE(103,36)
1170 POKE(104,69)
1180 OPEN B1,I,JOURNAL.DAT
1190 READ B1,A0
1200 A=USER(X)
1210 CLOSE #1
1220 GOSUB 120
1230 IF A<0 THEN PRINT "UN";
1240 PRINT "SUCCESSFUL BACKUP"
1245 IF A<0 THEN 1320
1250 PRINT "CURRENT JOURNAL ON B0"
1260 PRINT :PRINT "PUT LATEST O/L ON B1"
1270 INPUT "IS IT THERE",A0
1280 IF A0="" THEN 1270
1290 IF LEFT$(A0,1)<>"Y" THEN 1260
1300 IF F=2 THEN CHAIN 1,OPNT.BAS,20
1310 CHAIN 1,OPNT.BAS,10
1320 INPUT "TRY AGAIN",A0
1330 IF A0="" THEN 1320
1340 IF LEFT$(A0,1)= "Y" THEN 1180
1350 IF LEFT$(A0,1)= "N" THEN 1250
1360 GOTO 1230
2990 REM
2995 REM JOURNAL PRINT
2997 REM
3000 GOSUB 120
3010 PRINT "JOURNAL PRINT"
3020 PRINT :INPUT "DO YOU WANT ALL TRANSACTIONS",A0
3030 IF A0="" THEN 3020
3040 IF LEFT$(A0,1)= "Y" THEN T=1:GOTO 3160
3050 IF LEFT$(A0,1)<>"N" THEN 3020
3060 PRINT "TRANSACTIONS BY SEQ B"
3070 PRINT TAB(1);:INPUT "OR BY DATE ID OR O",A0
3080 IF A0="" THEN 3040
3090 IF LEFT$(A0,1)= "B" THEN T=2:GOTO 3120
3100 IF LEFT$(A0,1)= "O" THEN T=3:GOTO 3140
3110 GOTO 3040
3120 INPUT "BEGINNING SEQ #",B
3130 PRINT TAB(4);:INPUT "ENDING SEQ #",E:GOTO 3160
3140 INPUT "BEGINNING DATE (MMDD)",B
3150 PRINT TAB(4);:INPUT "ENDING DATE (MMDD)",E
3160 PRINT :INPUT "DATA OK",A0
3170 IF A0="" THEN 3140
3180 IF LEFT$(A0,1)<>"Y" THEN 3000
3190 OPEN R1,I,JOURNAL.DAT

```

```

3200 GOSUB 120:PRINT "ACCT BEG DATE EMBIT DEBIT"
3210 READ B1,P1,P2,P3,P4,P5
3215 IF EOF(1)=1 THEN 3500
3220 ON 1 GOTO 3280,3230,3260
3230 IF P2<8 THEN 3210
3240 IF P2<E THEN 3500
3250 GOTO 3280
3260 IF INT(P3/100)<8 THEN 3210
3270 IF INT(P3/100)>E THEN 3500
3280 A=STR$(P4)
3290 IF P4=INT(P4) THEN A=A+"."+"0"
3300 IF P4=10=INT(P4+10) THEN A=A+"0"
3310 B=STR$(INT(P3/100))
3320 IF P3=9999 THEN B=LEFT$(B,2)+"0"
3330 IF P3=100000 THEN B=LEFT$(B,1)+"0"
3340 PRINT P1;TAB(10-LEN(B));P2;
3350 PRINT TAB(16-LEN(B)-1);B;
3360 IF P4<0 THEN PRINT TAB(10+7);
3370 PRINT A
3380 IF ASC(P4)<32 THEN 3210
3390 PRINT P4
3400 GOTO 3210
3500 PRINT
3510 INPUT "BOME; PREED RETURN. ",A6
3520 CHAIN 0.MENU

```

```

0001 REM THIS IS THREE PROGRAMS -
0002 REM END YEAR, END PERIOD, TRIAL BALANCE
0003 STOP :END GPNT.BAS
0010 F=1:GOTO 100:REM END YEAR
0020 F=2:GOSUB 120:GOTO 100:REM END PERIOD
0030 F=3:GOTO 100:REM TRIAL BALANCE
0100 ON F GOTO 1000,1360,2000
0110 REM
0113 REM VARIOUS SUBROUTINES
0115 REM
0120 PRINT CHR$(116);CHR$(122);CHR$(0);CHR$(0);CHR$(0);
0130 RETURN
0300 A=STR$(A)
0310 IF A=INT(A) THEN A=A+"."+"0"
0320 IF A10=INT(A10) THEN A=A+"0"
0330 RETURN
0400 PRINT " ";B2;TAB(10);
0410 A=ABS(B4)
0420 GOSUB 300
0430 IF B4<0 THEN PRINT TAB(10+31);
0440 PRINT A6
0450 RETURN
0993 REM
0995 REM END OF YEAR
0997 REM
1000 GOSUB 120
1010 PRINT "I NEED SYSTEM DISC ON B0"
1020 INPUT "IS IT THERE",A6
1030 IF A6="" THEN 1020
1040 IF LEFT$(A6,1)<>"Y" THEN 1010
1050 OPEN B1,O.PRN.DAT
1060 READ B1,V1,V2,V3,V4,V5,V6,V7,V8,V9
1070 CLOSE B1
1080 GOSUB 120
1090 PRINT "PUT OLDEST O/L DISC ON B0"
1100 INPUT "IS IT THERE",A6
1110 IF A6="" THEN 1100
1120 IF LEFT$(A6,1)<>"Y" THEN 1090
1130 OPEN B1,O.OLN$1.BAT
1140 SCRATCH B1
1150 WRITE B1,01
1160 CLOSE B1
1170 OPEN B1,I.HIAB7FR.DAT
1180 OPEN B2,O.OLHASTER.AT
1190 SCRATCH B2
1200 READ B1,B1,02,03,03
1210 IF EOF(1)=1 THEN 1270
1220 IF B1<V4 THEN P=P+B2:G2=0:G010 1250
1230 IF B1>V6 THEN IF B1<V7 THEN P=P+G2:G2=0:G010 1250
1240 IF B1>V8 THEN G2=02-P
1250 WRITE B2,B1,02,03,03
1260 GOTO 1200
1270 CLOSE B1,B2
1280 GOSUB 120
1290 IF P>0 THEN PRINT "PROFIT FOR YEAR = $";
1300 IF P<0 THEN PRINT "LOSS FOR YEAR = $";
1310 A=ABS(P)
1320 GOSUB 300
1330 PRINT A6
1340 PRINT A6
1350 PRINT A6
1353 REM
1355 END OF PERIOD + END OF YEAR
1357 REM
1360 PRINT "DO YOU WANT TAPE BACKUP?"
1370 INPUT "FOR O/L",A6
1380 IF A6="" THEN 1360
1390 IF LEFT$(A6,1)<>"N" THEN 1510
1400 IF LEFT$(A6,1)<>"Y" THEN 1360
1410 IF B1=0 THEN OPEN B1,I.OLN$1.BAT
1415 IF B1=1 THEN OPEN B1,I.LHASTER.DAT
1420 READ B1,A
1430 POKE(103,34)
1440 POKE(104,72)

```

```

1450 A=UBERIX)
1460 CLOSE B1
1465 GOSUB 120
1470 IF A<0 THEN IF A<B THEN PRINT "UM";
1480 PRINT "SUCCESSFUL BACKUP OF FILE ";V+1
1490 IF A<0 THEN IF A<B THEN 1590
1500 INPUT "PRESS RETURN. ",A6
1510 GOSUB 120
1515 IF B=0 THEN B=1:GOTO 1410
1520 IF F=2 THEN 1460
1530 PRINT "YOUR LATEST O/L IS ON B0"
1540 PRINT "I NEED SYSTEM DISC ON B0"
1550 INPUT "IS IT THERE",A6
1560 IF A6="" THEN 1550
1570 IF LEFT$(A6,1)<>"Y" THEN 1540
1580 CHAIN 0.MENU
1590 INPUT "TRY AGAIN",A6
1593 IF A6="" THEN 1590
1595 IF LEFT$(A6,1)<>"Y" THEN 1410
1597 GOTO 1500
1600 OPEN B1,I.OLHIST.DAT
1610 SCRATCH B1
1620 A=0
1630 WRITE B1,A
1640 CLOSE B1
1650 GOSUB 120
1660 PRINT "YOUR LATEST O/L IS ON B1"
1670 PRINT "I NEED SYSTEM DISC ON B0"
1680 INPUT "IS IT THERE",A6
1690 IF A6="" THEN 1680
1700 IF LEFT$(A6,1)<>"Y" THEN 1670
1710 CHAIN 0.MENU
1993 REM
1995 REM TRIAL BALANCE
1997 REM
2000 GOSUB 120
2010 INPUT "DO YOU WANT HISTORY TO PRINT",A6
2020 IF A6="" THEN 2010
2030 IF LEFT$(A6,1)<>"Y" THEN B=2:GOTO 2040
2040 IF LEFT$(A6,1)<>"N" THEN 2010
2050 B=1
2060 OPEN B1,I.OLHASTER.DAT
2070 IF B=2 THEN OPEN B2,I.OLN$1.BAT
2080 GOSUB 120
2090 OPEN B3,O.PRN.DAT
2100 READ B3,V1,V2,V3,V4,V5,V6,V7,V8,V9
2110 CLOSE B3
2120 V=STR$(V1)
2130 IF V1<100000 THEN A=LEFT$(V,1)+"0"
2140 IF V1>99999 THEN A=LEFT$(V,2)+"0"
2150 V=A+"0"
2160 PRINT "TRIAL BALANCE";TAB(20);V
2170 PRINT
2180 READ B1,B1,02,03
2190 IF EOF(1)=1 THEN 2400
2195 IF B2=0 THEN 2180
2200 IF B1<V4 THEN B=B+B2:G010 2180
2210 IF B1>V6 THEN IF B1<V7 THEN B=B+G2:G010 2180
2220 IF B1>V8 THEN B=B+G2:G010 2250
2230 IF F1=0 THEN GOSUB 2400
2240 F1=1:G2=02
2250 A=ABS(B2)
2260 GOSUB 300
2270 PRINT B1;G2;
2280 IF B2=0 THEN IF B2>22-LEN(A6) THEN PRINT
2290 PRINT TAB(22-LEN(A6));
2300 IF B2<0 THEN PRINT TAB(32-LEN(A6));
2310 PRINT A6
2330 IF B1=1 THEN 2180
2340 IF EOF(2)=1 THEN 2180
2350 IF B1>B1 THEN PRINT:GOTO 2180
2360 IF B1=0 THEN AGGUP 400
2370 READ B2,B1,02,03,04
2380 GOTO 2340
2400 PRINT "SURPLUS";
2410 A=ABS(B)
2420 V=STR$(B)
2430 PRINT TAB(22-LEN(A6));
2440 IF B<0 THEN PRINT TAB(10+10);
2450 PRINT A6
2460 PRINT A6
2470 A=ABS(B)
2480 GOSUB 300
2490 PRINT TAB(15-LEN(A6));
2500 IF B<0 THEN PRINT TAB(10+10);
2510 PRINT A6
2520 PRINT
2540 RETURN
2600 A=ABS(B)
2610 GOSUB 300
2620 PRINT TAB(15-LEN(A6));A6;
2630 A=ABS(B)
2640 GOSUB 300
2650 PRINT TAB(25-LEN(A6));A6
2660 PRINT A6
2670 CLOSE B1
2680 IF B=2 THEN CLOSE B2
2690 INPUT "PRESS RETURN. ",A6
2700 CHAIN 0.MENU

```

To be continued

DUMPFILe and DUMPCMDs, Two User Command Routines for Mini-Flex

by

Robert L. Pigford
300 Wilson Road
Newark, Del., 19711

INTRODUCTION:

Although the authors of the Mini-Flex operating system at Technical Systems Consultants provided many very useful command subroutines for management of files stored on floppy disks they did not include routines for listing the contents of binary files. Sometimes the user wants to determine what the contents of such files may be without moving them into the memory locations from which they will operate. This can be done easily and quickly with either of the two command routines listed below.

Either routine is called from the user terminal by typing DUMPFILe, FILE SPEC or DUMPCMDs, FILE SPEC. Although the default extension is .BIN, binary files having extensions such as .CMD or .SYS can of course be processed by including the appropriate extension in the file specification. Both programs operate from the short memory block in DOS used by other short command routines. Each program can be used for output on either terminal or printer.

DUMPFILe, FOR OUTPUT OF BINARY AND ASCII DATA:

This program displays the data bytes in the file both in hexadecimal and alphanumeric form in lines of sixteen bytes on the terminal. The address of the byte at the first of each line is printed first, followed by the hex data. Then the ASCII equivalent of each of the sixteen bytes is displayed in a second line, if it exists. Otherwise a period is displayed.

The program itself is shown in Listing 1. It is similar to LOCATE(1), which used portions of T.S.C.'s DCS program for opening a file on disk, reading single bytes from it, and checking for disk errors.

The heart of the program is the subroutine called LOADER. It reads bytes from the disk, computes their addresses, and outputs them to the terminal. Some lines do not extend the full length if the program comes to the end of a data block on the disk first. Thus, one may see at a glance in the output whether the data come from one block or the next. At the end of the program the starting address of the program is displayed and the program returns to the disk operating system.

After finding from the disk the beginning address of a data block and the number of bytes in that block, LOADER outputs the beginning address and reads into a buffer either sixteen bytes or the number left in the block if that is smaller. Next the buffer contents are displayed and, afterward, the same contents are

given in ASCII form. At several points in the program control returns to the beginning of LOADER if a new block needs to be read from disk.

DUMPCMDS, FOR OUTPUT OF DATA IN STANDARD COMMAND FORMAT:

The program is shown in Listing 2 and is similar to the previous program. The logic of its LOADER subroutine is different, however, because each op-code read from the disk must be classified to determine whether it represents a single-byte command or one of two or three bytes. This is accomplished by the program segment called INSTYP which was published by N.J.Thompson(2). The number of bytes in each command is stored in SAVEB, which is referred to several times to determine whether a command has been finished or not. Another counter, BCNTR, stores the number of bytes left in each block of data on the disk. Whenever it becomes zero, even in the middle of a command, the program returns to read a new block on the disk but the returns are marked by setting either of two flags, FLAG1 or FLAG2. These are tested to determine whether a new data byte just read from the disk is either the second or the third byte of an incomplete command. The program returns to the appropriate point to finish output of the whole command.

Here is a sample of the output:

```
760C 26 00
760E BD 71 1E
7611 BD 75 2A
7614 25 07
7616 7C 70 B3
7619 8D 0D
761B 20 EF
761D F6 70 B3
```

CONCLUSIONS:

I hope you will find these programs useful, as I have. They can be put to use easily either by assembly on your own computer or by introducing the machine language from Listing 1 or Listing 2 into your equipment. Anyone who wants a copy on disk should send his own disk to me plus fifty cents for return postage.

REFERENCES:

- 1.) Pigford, R.L., "LOCATE, a Utility Command Program for Flex", '68' Micro Journal, March-April, 1979, p.26
- 2.) Thompson, N.J., Byte, Oct., 1976, p.99

```
1          NAM      DUMPFIL
2          *
3          *****
4          * DUMPFIL IS A UTILITY COMMAND *
5          * PROGRAM FOR FLEX. IT DISPLAYS *
6          * ON THE TERMINAL OR THE PRINTER *
7          * THE CONTENTS OF A BINARY FILE *
8          * IN HEX AND IN ASCII. VERSION 2 *
9          * 7-1-79, BY R.L.PIGFORD      *
10         *****#
11         *
```

12				* SYSTEM EQUATES	
13				*	
14	7118			PSTRNG EQU \$7118	CR/LF & print string in Flex-1
15	711E			PCRLF EQU \$711E	CR/LF
16	7139			OUTHEX EQU \$7139	output hex byte
17	758A			OPENFI EQU \$758A	get file spec. and open for read
18	7525			FMS1 EQU \$7525	read data from disk and check
19	71BD			ERFOR EQU \$71BD	
20	7103			WARMS EQU \$7103	return to Flex-1
21	7112			PUTCHR EQU \$7112	output character in Flex-1
22	70B3			SCRATCH EQU \$70B3	
23	709F			TADDR EQU \$709F	transfer address
24				*	
25				*	
26	7600			ORG \$7600	assemble in user command space
27	7600	20 00		START BRA INIT1	
28	7602	02		VN FCB 2	version 2
29	7603			CHAR RMB 1	store hex byte for output
30	7604			SAVEB RMB 1	no. bytes left in block
31	7605			CNTR RMB 1	no. bytes in one line output
32	7606			CNTRL RMB 1	similar for character output
33	7607			XSAVE RMB 2	address counter
34	7609			BUFADR RMB 2	buffer address counter
35	760B	BD 71 1E		INIT1 JSR PCRLF	CR/LF
36	760E	36 00		INIT LDA A #0	
37	7610	BD 75 8A		JSR OPENFI	get file spec.; open for read
38	7613	25 07		BCS RETURN	return to Flex if through
39	7615	7C 70 B3		INC SCRATCH	
40	7618	8D 0D		BSR LOADER	go to read file and output
41	761A	20 F2		BRA INIT	
42	761C	F6 70 B3	RETURN	LDA B SCRATCH	
43	761F	27 03		BEQ JUMP	
44	7621	7E 71 03		JMP WARMS	return to Flex
45	7624	7E 71 BD	JUMP	JMP ERFOR	
46			*		
47	7627	BD 75 25	LOADER	JSR FMS1	get data byte from file
48	762A	81 02		CMP A #2	start of new record?
49	762C	27 1E		BEQ LDR2	branch if yes
50	762E	21 16		CMP A #16	transfer addr. follows?
51	7630	26 F5		BNE LOADER	return for another byte if no
52	7632	BD 75 25		JSR FMS1	get program entry addr.
53	7635	B7 70 9F		STA A TADDR	store it
54	7638	BD 75 25		JSR FMS1	
55	763B	B7 70 A0		STA A TADDR+1	
56	763E	CE 77 22		LDX #MSG3	get start-addr. message
57	7641	BD 71 18		JSR PSTRNG	output it
58	7644	CE 70 9F		LDX #TADDR	output address for
59	7647	BD 77 12		JSR OUT4HS	program entry
60	764A	20 DB		BR LOADER	return
61	764C	BD 75 25	LDR2	JSR FMS1	get addr. of data record from
62	764F	B7 76 07		STA A XSAVE	file; save in XSAVE
63	7652	BD 75 25		JSR FMS1	
64	7655	B7 76 08		STA A XSAVE+1	
65	7658	BD 75 25	LDR5	JSR FMS1	get no. bytes in record block
66	765B	16		TAB	save in B-register
67	765C	B7 76 04		STA A SAVEB	save also in SAVEB
68	765F	27 C6		BEQ LOADER	return if record is blank
69	7661	BD 71 1E	LDR8	JSP PCRLF	CR?LF
70	7664	CE 76 07		LDX #XSAVE	point to addr. counter
71	7667	BD 77 12		JSP OUT4HS	output address

72	766A	BD	77	1C	JSP	OUTS	plus two extra spaces
73	766D	BD	77	1C	JSR	OUTS	
74	7670	CE	77	2E	LDX	#BUFTOP	initialize buffer pointer
75	7673	FF	76	09	STX	BUFADR	
76	7676	C1	10		CMP B	#16	compare no. bytes in block
77	7678	23	07		BLS	LEQ	with 16; branch if too few
78	767A	86	10		LDA A	#16	to fill buffer
79	767C	B7	76	05	STA A	CNTR	initialize line cntr. = 16
80	767F	20	03		BRA	LDR3	
81	7681	F7	76	05	STA B	CNTR	initialize cntr. = no. bytes left
82	7684	BD	75	25	JSR	FMS1	get data byte from file
83	7687	FE	76	09	LDX	BUFADR	
84	768A	A7	00		STA A	0,X	store it in buffer
85	768C	08			INX		
86	768D	FF	76	09	STX	BUFADR	save buffer pointer
87	7690	7A	76	05	DEC	CNTR	reduce line counter
88	7693	27	05		BEQ	LDR6	buffer filled?
89	7695	5A			DEC B		no; return for new byte
90	7696	26	EC		BNE	LDR3	
91	7698	20	8D		BRA	LOADER	return for another block
92	769A	37			PSH B		save B-reg. on stack
93	769B	F6	76	04	LDA B	SAVEB	get no. bytes left in block
94	769E	C1	10		CMP B	#16	
95	76A0	23	0A		BLS	LEQ1	
96	76A2	86	10		LDA A	#16	
97	76A4	B7	76	05	STA A	CNTR	initialize both line counters = 16
98	76A7	B7	76	06	STA A	CNTR1	
99	76AA	20	06		BRA	GT	
100	76AC	F7	76	05	STA B	CNTR	initialize both counters =
101	76AF	F7	76	06	STA B	CNTR1	no. bytes left in block
102	76B2	CE	77	2E	LDX	#BUFTOP	point to beginning of buffer
103	76B5	FF	76	09	STX	BUFADR	
104	76B8	B6	76	04	LDA A	SAVEB	subtract no. bytes in full
105	76BB	80	10		SUB A	#16	line from no. left in block
106	76BD	B7	76	04	STA A	SAVEB	save the result
107	76C0	FE	76	09	LDX	BUFADR	get byte from buffer
108	76C3	A6	00		LDA A	0,X	
109	76C5	08			INX		increment buffer address
110	76C6	FF	76	09	STX	BUFADR	save it
111	76C9	B7	76	03	STA A	CHAR	store character for now
112	76CC	CE	76	03	LDX	#CHAR	point to character
113	76CF	8D	45		BSR	OUT2HS	output it + space
114	76D1	FE	76	07	LDX	XSAVE	increment addr. counter
115	76D4	08			INX		
116	76D5	FF	76	07	STX	XSAVE	save it
117	76D8	7A	76	05	DEC	CNTR	decrement line counter
118	76DB	26	E3		BNE	LDR9	return if more in line
119	76DD	CE	77	2E	LDX	#BUFTOP	point to buffer beginning
120	76E0	FF	76	09	STX	BUFADR	
121	76E3	BD	71	1E	JSR	PCRLF	CR/LF
122	76E6	C6	08		LDA B	#8	
123	76E8	8D	32		BSR	OUTS	output 8 spaces
124	76EA	5A			DEC B		
125	76EB	26	FB		BNE	A1	
126	76ED	A6	00		LDA A	0,X	get a character from buffer
127	76EF	81	5A		CMP A	#5A	is it above 'Z'?
128	76F1	22	06		BHI	A3	
129	76F3	81	1F		CMP A	#1F	is it below SP?
130	76F5	23	02		BLS	A3	
131	76F7	20	02		BRA	A4	

132	76F9	86	2E	A3	LDA A	#\$2E	substitute period, '.
133	76FB	BD	71 12	A4	JSR	PUTCHR	output char. in ASCII
134	76FE	0E			INX		increment buffer addr.
135	76FF	8D	1E		BSR	OUTS	print two extra spaces
136	7701	8D	19		BSR	OUTS	
137	7703	7A	76 06		DEC	CNTR1	reduce line counter
138	7706	26	E5		BNE	A2	return if more in buffer
139	7708	33			PUL B		retrieve no. bytes left in
140	7709	5A			DEC B		block; decrement it and
141	770A	26	03		BNE	LDR10	test if none left
142	770C	7E	76 27		JMP	LOADER	return to LOADER if none
143	770F	7E	76 61	LDR10	JMP	LDR8	return for new output line
144				*			
145	7712	BD	71 39	OUT4HS	JSR	OUTHEX	start here to output 4 hex
146	7715	0E			INX		and SP
147	7716	BD	71 39	OUT2HS	JSR	OUTHEX	start here to output 2 hex
148	7719	8D	01		BSR	OUTS	and SP
149	771B	39			RTS		
150				*			
151	771C	86	20	OUTS	LDA A	#\$20	output space
152	771E	BD	71 12		JSR	PUTCHR	
153	7721	39			RTS		
154				*			
155	7722	0A	0D	MSG3	FDB	\$0A0D	
156	7724	53			FCC	/START AT /	
157	772D	04			FCB	4	
158	772E			BUFTOP	RMB	16	buffer storage for one line
159					END	START	

NO ERROR(S) DETECTED

1				NAM	DUMPCMDS	
2				*		
3				*****		
4				*DUMPCMDS IS A UTILITY COMMAND *		
5				* PROGRAM FOR FLEX. IT DISPLAYS*		
6				* ON THE TERMINAL OR PRINTER *		
7				* THE CONTENTS OF A BINARY FILE *		
8				* IN STANDARD COMMAND FORMAT *		
9				* WITHOUT LOADING THE FILE INTO *		
10				* RAM. VERSION 1. 5-15-79 *		
11				* BY R. L. PIGFORD *		
12				*****		
13				*		
14				* SYSTEM EQUATES		
15				*		
16	7118			PSTRNG	EQU	\$7118 CR/LF & print string in Flex-1
17	711E			PCRLF	EQU	\$711E CR/LF only
18	7139			OUTHEX	EQU	\$7139 output hex byte
19	7112			PUTCHR	EQU	\$7112 output character
20	758A			OPENFI	EQU	\$758A get file spec. and open for read
21	7525			FMS1	EQU	\$7525 read data from file and check
22	71BD			ERROR	EQU	\$71BD
23	7103			WARMS	EQU	\$7103 return to Flex-1
24	70B3			SCRATCH	EQU	\$70B3
25	709F			TADDR	EQU	\$709F transfer address
26				*		
27				*		
28	7600			ORG	\$7600	assemble in DOS user command space
29	7600	20	0A	START	BRA	INIT
30	7602	01		VN	FCB	1 version

31	7603		SAVEX	RMB	2	command addr. counter
32	7605		BCNTR	RMB	1	no. of bytes counter
33	7606		SAVEB	RMB	1	no of bytes in command
34	7607		FLAG1	RMB	1	indicates incomplete 2-byte cmd.
35	7608		FLAG2	RMB	1	indicates incomplete 3-byte cmd.
36	7609		BYTE1	RMB	1	op-code of command
37	760A		BYTE2	RMB	1	first operand byte
38	760B		BYTE3	RMB	1	second operand byte
39			*			
40	760C	36 00	INIT	LDA A	#0	
41	760E	BD 71 1E		JSR	PCRLF	CR/LF
42	7611	BD 75 8A		JSR	OPENFI	get file spec.; open for read
43	7614	25 07		BCS	RETURN	
44	7616	7C 70 B3		INC	SCRATCH	
45	7619	8D 0D		BSR	LOAD2	go to routine to read file and
46	761B	20 EF		BFA	INIT	output commands
47	761D	FG 70 B3	RETURN	LDA B	SCRATCH	
48	7620	27 03		BEQ	JUMP	
49	7622	7E 71 03		JMP	WARMS	return to Flex-1
50	7625	7E 71 3D	JUMP	JMP	ERROR	
51			*			
52			*			
53	7628	7F 76 07	LOAD2	CLR	FLAG1	clear flags to begin command
54	762B	7F 76 08		CLR	FLAG2	
55	762E	BD 75 25	LOADER	JSP	FMS1	get byte from file
56	7631	81 02		CMP A	#2	is it new-record indicator?
57	7633	27 1E		BEQ	LDR2	
58	7635	81 1C		CMP A	#16	is it trans.-addr. indicator?
59	7637	26 15		BNE	LOADER	return for another byte
60	7639	BD 75 25		JSR	FMS1	get program entry address
61	763C	B7 70 9F		STA A	TADDR	store it
62	763F	BD 75 25		JSR	FMS1	
63	7642	B7 70 A0		STA A	TADDR+1	
64	7645	CE 77 23		LDX	#MSG3	point to begin-addr. message
65	7648	BD 71 18		JSR	PSTRNG	print it
66	764B	CE 70 9F		LDX	#TADDR	point to transfer address
67	764E	BD 77 01		JSR	OUT4HS	print it
68	7651	20 DE	LOAD1	BFA	LOADER	return for another byte and
69	7653	BD 75 25	LDR2	JSR	FMS1	get 2-byte begin addr.
70	7656	B7 76 03		STA A	SAVEX	from file
71	7659	BD 75 25		JSR	FMS1	
72	765C	B7 76 04		STA A	SAVEX+1	
73	765F	BD 75 25	LDR5	JSP	FMS1	get no. bytes in record block
74	7662	B7 76 05		STA A	BCNTR	store for future use
75	7665	27 C7		BEQ	LOADER	return if block is blank
76	7667	BD 75 25	EPSILON	JSP	FMS1	get data byte from file and
77	766A	7A 76 05		DEC	BCNTR	decrement counter
78	766D	7D 76 07		TST	FLAG1	test for incomplete 2-byte cmd.
79	7670	26 53		BNE	GAMMA1	return to 2-byte output
80	7672	7D 76 08		TST	FLAG2	test for incomplete 3-byte cmd.
81	7675	26 72		BNE	DELTA1	return to 3-byte output
82	7677	B7 76 09		STA A	BYTE1	store byte as op-code
83	767A	5F		CLR B		clear no. bytes in command
84	767B	CE 76 03		LDX	#SAVEX	point to addr. counter
85	767E	BD 77 01		JSR	OUT4HS	output addr. of cmd. & SP
86	7681	BD 77 15		JSR	OUTS	plus a second space
87	7684	CE 76 09		LDX	#BYTE1	point to op-code
88	7687	8D 7C		BSR	OUT2HS	output it + space
89	7689	BD 77 1B		JSR	INCSAV	increment cmd. addr. counter
90	768C	B6 76 09		LDA A	BYTE1	get op-code again

91	768F	81	8C	INSTYP	CMP A	#\$8C	test it for 1, 2, or 3 bytes
92	7691	27	13		BEQ	THREEB	in the command
93	7693	81	3E		CMP A	#\$8E	
94	7695	27	14		BEQ	THREEB	
95	7697	81	CE		CMP A	#\$CE	
96	7699	27	10		BEQ	THREEB	
97	769B	84	F0		AND A	#\$F0	
98	769D	81	20		CMP A	#\$20	
99	769F	27	0B		BEQ	TWOB	
100	76A1	81	60		CMP A	#\$60	
101	76A3	25	08		ECS	ONEB	
102	76A5	84	30		AND A	#\$30	
103	76A7	81	30		CMP A	#\$30	
104	76A9	26	01		BNE	TWOB	
105	76AB	5C		THREEB	INC B		B-reg. will contain no. bytes
106	76AC	5C		TWOB	INC B		in the command
107	76AD	F7	76 06	ONEB	STA B	SAVEB	store no. cmd. bytes in SAVEB
108	76BD	26	09		BNE	MU	branch if 1- or 2-byte cmd.
109	76B2	BD	71 1E	END1	JSR	PCRLF	
110	76B5	8D	54		BSR	TESTB	test no. bytes left in record
111	76B7	25	93		BCS	LOAD1	branch if all gone
112	76B9	20	AC		BRA	EPSILON	get new data from disk
113	76BB	8D	4E	MU	BSR	TESTB	test no. bytes left in block
114	76BD	24	0B		BCC	GAMMA	branch if bytes are left
115	76BF	7C	76 07		INC	FLAG1	set flag to show incomplete
116	76C2	7E	76 2E		JMP	LOADER	2-byte cmd.; get data from
117	76C5	7F	76 07	GAMMA1	CLR	FLAG1	next block; clear flag upon
118	76C8	20	03		BRA	GAMMA2	return
119	76CA	BD	75 25	GAMMA	JSR	FMS1	get new byte from file
120	76CD	B7	76 0A	GAMMA2	STA A	BYTE2	store it as first operand byte
121	76D0	7A	76 05		DEC	BCNTR	reduce bytes counter
122	76D3	CE	76 0A		LDX	#BYTE2	point to first operand
123	76D6	8D	2D		BSF	OUT2HS	output it
124	76D8	8D	41		BSR	INCSAV	increment addr. counter
125	76DA	7A	76 06		DEC	SAVEB	reduce no. bytes in command
126	76DD	27	D3		BEQ	END1	return if command complete
127	76DF	8D	2A		BSF	TESTB	test for bytes left in block
128	76E1	24	0B		BCC	DELTA	branch if some are left
129	76E3	7C	76 08		INC	FLAG2	set flag to show need byte 3
130	76E6	7E	76 2E		JMP	LOADER	return for a new record block
131	76E9	7F	76 02	DELTA1	CLR	FLAG2	clear flag upon return
132	76EC	20	03		BRA	DELTA2	
133	76EE	BD	75 25	DELTA	JSR	FMS1	get second operand byte
134	76F1	B7	76 0B	DELTA2	STA A	BYTE3	store it
135	76F4	7A	76 05		DEC	BCNTR	reduce bytes counter
136	76F7	CE	76 0B		LDX	#BYTE3	point to second operand
137	76FA	8D	09		BSR	OUT2HS	output it
138	76FC	8D	1D		BSR	INCSAV	increment addr. counter
139	76FE	7E	76 B2		JMP	END1	return; command finished
140				*			
141	7701	BD	71 39	OUT4HS	JSR	OUTHEX	output 2 hex bytes
142	7704	08			INX		
143	7705	BD	71 39	OUT2HS	JSR	OUTHEX	output 1 hex byte
144	7708	8D	0B		BSR	OUTS	output space
145	770A	39			RTS		
146				*			
147	770B	7D	76 05	TESTB	TST	BCNTR	test no. bytes left in block
148	770E	27	03		BEQ	SETC	
149	7710	0C			CLC		clear carry if bytes are left
150	7711	20	01		BRA	RETN	

```

151 7713 0D      SETC      SEC      set carry if all bytes used
152 7714 39      RETN      RTS
153
154 7715 86 20    OUTS      LDA A    #20    output a space
155 7717 BD 71 12 JSR      PUTCHP
156 771A 39      RTS
157
158 771B FE 76 03 INCSAV    LDX      SAVEX  increment address counter
159 771E 0C      INX
160 771F FF 76 03 STX      SAVEX  store it
161 7722 39      RTS      return
162
163 7723 53      MSG3      FCC      /STARTING ADDRESS = /
164 7736 04      FCB      4
165
166                      END      START

```

NO ERROR(S) DETECTED

The following programs enable data files and BASIC programs to be transferred from a PDP-11 to an SWTPc 6809. POP-11 BASIC is so similar to SWTPc BASIC that very little effort is required to run the POP-11 programs on the 6809.

The programs supplied on this disk have port addresses set for the S/09 computer. An RS-232 serial interface from the POP-11 is plugged into the S/09 port 0A. (The YOU control terminal should be in port 0B as usual). The line between the POP-11 and the 6809 should be configured as follows:

```

PIN 2 to PIN 3
PIN 3 to PIN 2
PIN 7 to PIN 7

```

T06809.BAS This program is for the PDP-11 and should be started first. It informs the operator when the 6809 program is to be started. It is essential that the PDP-11 operates in SIMPLEX mode (no echo) on the interface line.

FROMPOP.BAS This is an XBASIC program for the 6809. It receives the file transmitted from the PDP-11.

FROMPOP.BAC "Compiled" version of above program.

OPENPDP.CMD Used by the program FROMPOP to switch the echo off and to direct XBASIC to port 0A. Source code supplied in case port modifications are required.

CLOSEPDP.CMD Used by the program FROMPOP to switch the echo back on and to direct XBASIC back to port 0B. Source code supplied in case port modifications are required.

```

1 REM PDP-11 PROGRAM TO TRANSMIT FILES TO SWTPc 6809
2 REM -----
3 REM
10 MS="GO"
30 PRINT "PDP-11 FILENAME";
40 INPUT #0,FS
50 IF FS="" THEN STOP
70 OPEN FS FOR INPUT AS FILE 1
80 L=0: N=0
85 OPEN "KB1:" AS FILE 2%
90 PRINT "START 6809 PROGRAM NOW"
95 ON ERROR GOTO 900
100 INPUTLINE #2%,AS: AS=LEFT(AS,2)
105 IF AS=MS GOTO 150
110 INPUTLINE #2%,BS: BS=LEFT(BS,2)
115 IF BS=MS GOTO 150
120 INPUTLINE #2%,CS: CS=LEFT(CS,2)
125 IF CS=MS GOTO 150
130 INPUTLINE #2%,DS: DS=LEFT(DS,2)
140 PRINT "NO GO":PRINT "AS=" AS "BS=" BS "CS=" CS "DS=" DS
145 STOP
150 FOR I=1 TO 50:NEXT I
160 INPUTLINE #1,LS
170 PRINT #2%,"Z";LS;: PRINT CHR$(13);
180 L=L+1: N=N+LEN(LS)-2
190 GOTO 100
200 PRINT #2%,"***"

```

```

210 PRINT L "LINES" N "CHARACTERS"
230 CLOSE 1,2%
250 GOTO 30
900 IF ERL=160 AND ERR=11 THEN RESUME 200
910 ON ERROR GOTO 0
920 END

```

```

1 REM PROGRAM TO RECEIVE FILES FROM POP-11
2 REM -----
3 REM
100 PRINT "NAME OF FILE TO BE CREATED? "; INPUTLINE #0,FS: PRINT
105 IF FS="" THEN STOP
110 ON ERROR GOTO 900
120 OPEN OLD FS AS 1: REM CHECK IF FILE ALREADY EXISTS
125 INPUTLINE #1,XS
130 CLOSE 1
135 IF E%=4 THEN 200
140 PRINT "DELETE EXISTING FILE? "; RS=INCHS(0): PRINT
150 IF RS="N" THEN 100
160 IF RS<>"Y" THEN 140
200 OPEN NEW FS AS 1
210 PRINT "COMMENCING TRANSFER..."
220 EXEC,"OPENPDP"
230 LC=0: CC=0
300 PRINT "GO"
310 INPUTLINE #0,LS
325 IF LS="***" THEN 400
330 TS=MIOS(LS,2): REM AVOID LOSS OF LEADING SPACES
350 PRINT #1,TS
360 IS=LEN(TS)
370 LC=LC+1
380 CC=CC+IS
390 GOTO 300
400 EXEC,"CLOSEPDP"
410 CLOSE 1
420 PRINT LC;" LINES"
430 PRINT CC;" CHARACTERS"
450 GOTO 100
900 IF ERL=125 AND (ERR=8 OR ERR=4) THEN E%=ERR: RESUME 130
910 EXEC,"CLOSEPDP"
920 ON ERROR GOTO 0

```

* OPEN POP INTERFACE

```

*
REENTER EQU %CD06
INCH EQU %CD09
INCH2 EQU %CD0C
OUTCH EQU %CD0F
OUTCH2 EQU %CD12
CNTRL EQU %E000 ACIA CONTROL REGISTER
DATA EQU %E001 ACIA DATA REGISTER

```

* LOAD ACIA DRIVER ROUTINES INTO PRINT.SYS SPACE

```

ORG %CCCO
INV RMB 2
IN2V RMB 2
OUTV RMB 2
OUT2V RMB 2

```

```

ICHR LDA CNTRL INPUT CHARACTER ROUTINE
ANDA #S01
BEQ ICHAR WAIT FOR CHARACTER
LDA DATA LOAD CHARACTER
ANDA #S7F
RTS

```

```

OCHAR PSHS #S02 OUTPUT CHARACTER ROUTINE
OCHAR1 LDA CNTRL
ANDA #S02
BEQ OCHAR1 WAIT UNTIL ACIA READY
PULS #S02
STA DATA OUTPUT CHARACTER
RTS

```

```
ORG $C100
```

```

START BRA DPOP1
VN FCB 1 VERSION NUMBER

```

```

DPOP1 LDD INCH+1
STD INV
LDD #ICHR
STD INCH+1 RE-ASSIGN INCH
LDD INCH+1
STD IN2V
LDD #ICHR
STD INCH+1 RE-ASSIGN INCH2
LDD OUTCH+1
STD OUTV
LDD #OCHAR
STD OUTCH+1 RE-ASSIGN OUTCH
LDD OUTCH+1
STD OUT2V
LDD #OCHAR
STD OUTCH+1 RE-ASSIGN OUTCH2
LDA #S13
STA CNTRL INITIALISE ACIA
LDA #S11
STA CNTRL
JMP REENTER
END START

```

```

*
* CLOSE PDP INTERFACE
*

```

```

REENTER EQU $C006
INCH EQU $C009
INCH2 EQU $C00C
OUTCH EQU $C00F
OUTCH2 EQU $C012
INV EQU $CCCC
IN2V EQU $CCCC
OUTV EQU $CCCC
OUT2V EQU $CCCC

```

```
ORG $C100
```

```

START BRA CPOP1
VN FCB 1 VERSION NUMBER

```

```

CPOP1 LDD INV
STD INCH+1 RESET INCH
LDD IN2V
STD INCH2+1 RESET INCH2
LDD OUTV
STD OUTCH+1 RESET OUTCH
LDD OUT2V
STD OUTCH2+1 RESET OUTCH2
JMP REENTER
END START

```

WORKING WITH TSC'S RANDOM FILES
By Jim Schaler
(courtesy of INTERFACE AGE)

"It's the best there is for the 6800" is one way someone recently described the neat of BASICS TSC has been releasing since mid-spring of 1979. They are fast, accurate, easy to use and work without a hitch. There is no secret to the fact that TSC's forte is in software. They've made the 6800 user the envy of many S-100 bus users. But TSC's documentation is a little unusual. Instead of giving the user an endless list of dos and don'ts, TSC tells it like it is and presumes the reader understands the theory and shop talk. If you have a Masters in English Literature with a minor in Computer Science, TSC's documentation will be a breeze.

That lets me out. If it isn't presented as a simple rule with a program to detail the rule, I'm lost. The rules and sample programs TSC presents in the documentation for their FLEX-216 BASIC and Extended BASIC are pretty limited. If you have not had prior experience working with random files you may

find your self trying to figure out what is going on. Based on the old saying that a program is worth a thousand words I have adapted a sample program from Lon Poole's PAYROLL WITH COST ACCOUNTING (Osborne & Associates) showing how the TSC random files work. Since the documentation fails to include such a sample program, many other TSC BASIC users may find this example helpful, or at least readers expert in TSC random files may offer suggestions and alterations to what is being presented.

```

10 REM PROLOG
20 OPEN "1.GIOP010.0AT" AS B
30 GET B, RECORD 1
40 GOSUB 620
50 REM CONVERT STRINGS TO INTEGER AND FLOATING POINT
60 G1=CVT$(G1);G2=CVT$(G2);
70 G3=CVT$(G3);G4=CVT$(G4);
80 G5=CVT$(G5);G6=CVT$(G6);
90 X=C $(X);Y=CVT$(Y);Z=CVT$(Z);
100 GOSUB 540
110 REM DISPLAY MASK
120 PRINT TAB(10);"General Information File Maintenance"
130 PRINT
140 PRINT "1. Company Number";TAB(20);G1;
150 PRINT TAB(30);"10. Today's Date";TAB(50);M$
160 PRINT "2. Next Check Number";TAB(20);G2;
170 PRINT TAB(30);"11. Period Start";TAB(50);N$
180 PRINT "3. O. T. Rate";TAB(20);G3;
190 PRINT TAB(30);"12. Period Ending";TAB(50);G$
200 PRINT "4. Hourly Rate";TAB(20);G4;
210 PRINT TAB(30);"13. Payroll Number";TAB(50);G5$
220 PRINT TAB(30);"14. Day Number";TAB(50);G6$
230 PRINT "5. Company Name";TAB(20);A$
240 PRINT "6. Addr";TAB(20);B$
250 PRINT "7. ";TAB(20);C$
260 PRINT "8. ";TAB(20);D$
270 PRINT "9. Fed/State Nos.";TAB(20);E$
280 PRINT
290 PRINT "Enter Field No. to Change ('0' to Print; '-1' to Exit)";
300 INPUT S$
301 IF S$=0 THEN OPEN "0.PRINT" AS 0:GOTO 1000
310 IF S$=-1 THEN STOP
320 GOSUB 540:PRINT:PRINT
330 GOSUB 620
340 ON S$ GOTO 350,360,370,380,390,400,410,420,430,440,450,460,470,480
350 INPUT "Company Number",X$;L$ G1=CVT$(X$);GOTO 490
360 INPUT "Next Check Number",X$;L$ G2=CVT$(X$);GOTO 490
370 INPUT "O. T. Rate",Y$;L$ G3=CVT$(Y$);GOTO 490
380 INPUT "Hourly Rate",Y$;L$ G4=CVT$(Y$);GOTO 490
390 INPUT "Company Name",C$;L$ A$=C$;GOTO 490
400 INPUT "Addr line 1",A$;L$ B$=A$;GOTO 490
410 INPUT "Addr line 2",A$;L$ C$=A$;GOTO 490
420 INPUT "Addr line 3",A$;L$ D$=A$;GOTO 490
430 INPUT "Fed/State Nos.",F$;L$ E$=F$;GOTO 490
440 INPUT "Today's Date",D$;L$ M$=D$;GOTO 490
450 INPUT "Period Start",P$;L$ N$=P$;GOTO 490
460 INPUT "Period End",P$;L$ G$=P$;GOTO 490
470 INPUT "Payroll No.",P$;L$ G5$=CVT$(P$);GOTO 490
480 INPUT "Day Number",D$;L$ G6$=CVT$(D$);GOTO 490
490 PRINT:INPUT "Additional Changes, [ Y ] or [ N ] ",Z$
500 IF Z$="Y" THEN 100
510 REM WRITE TO DISK
520 PUT B, RECORD 1
530 END
540 REM ROUTINE TO CLEAR CT-64 CRT
550 FOR X$=1 TO 3:PRINT CHR$(26);NEXT X$
560 PRINT CHR$(2):RETURN
570 REM EXIT ROUTINE
580 FOR X$=1 TO 5:PRINT:NEXT X$
590 PRINT TAB(17);"--> G/1 PM Loading MZMU"
610 CHAIN "MENU.BAS":END
620 REM FIELD ALLOCATIONS
630 FIELD B,2ASG1$,2ASG2$,2ASG3$,2ASG4$,2ASAB$,2ASB$,2ASAC$,2ASAD$,2ASAE$,2ASAF$,
9ASNS$,9ASOS$,2ASOS$,2ASG6$,2ASX$,2ASY$,2ASZ$
640 RETURN
1000 REM ROUTINE TO PRINT GENERAL INFORMATION REPORT
1001 REM GOTO 100
1010 REM COL LENGTH = 132
(Notes: Print Routine not included in this sample program)

```

TRAP

Lee Johnson
2745 Northview Dr
Marion, IN 52302

TRAP is an assembly level program for loading program object files into system memory and returning control to the user before program execution begins. It is written for 6800 systems employing the FLEX (TM) disk operating system.

The FLEX disk operating system already has GET so why do we need another loader you say? GET is fine for loading programs without file specifications, but for those requiring file specification information it is not very useful. After GET loads the target program it returns control to FLEX and it remains an exercise for the user to figure out how to get the necessary file specification into the input line buffer. This exercise can be accomplished, but it can be a frustrating one indeed. TRAP solves this problem making the debug task for programs with file specifications easy and painless. Although TRAP was written to aid in debugging programs with file specifications, it can also be used for programs without them.

PROGRAM DESCRIPTION

TRAP makes use of several of the routines which reside in the FLEX repertoire of subroutines. It gets target file specifications from the input line buffer, loads the target program into memory, sets up the stack as though a software interrupt had occurred, and transfers control to the monitor software interrupt entry point. In my system TRAP is OR'd at location 06000, although this can be anywhere just as

WE HAVE A 6809 FOR YOU

POWER SUPPLY

Modular plug-in construction with computer grade filters and a 25 AMP rectifier bridge. Blower fan is standard equipment. All connections to the power line are beneath the safety shield.

INTERFACE

Convenient serial or parallel I/O cards have DB-25 connectors mounted directly on the circuit board. Up to 16 interface devices may be installed on the address decoded I/O bus. Programming strips are provided for input and output baud rate selection on each port. All outputs are fully buffered.

PROCESSOR

The world's most powerful eight-bit processor, the Motorola MC6809, plus 2K byte monitor ROM that is 2716 EPROM compatible and full buffering on all output lines. Built-in multiuser capability, just add I/O cards to operate a multi-terminal system.

CABINET

Rugged 1/8 inch alloy aluminum base plate combined with a solid 1/8 inch alloy aluminum cover for unsurpassed protection. All interior metal is conversion coated. The cover is finished with a super tough textured epoxy.

MEMORY— You can purchase the computer with either 8K bytes of RAM memory (expandable to 56K), or with the full 56K. The efficient, cool running dynamic memory used in this system is designed and manufactured for us by "Motorola Memory Systems Inc."

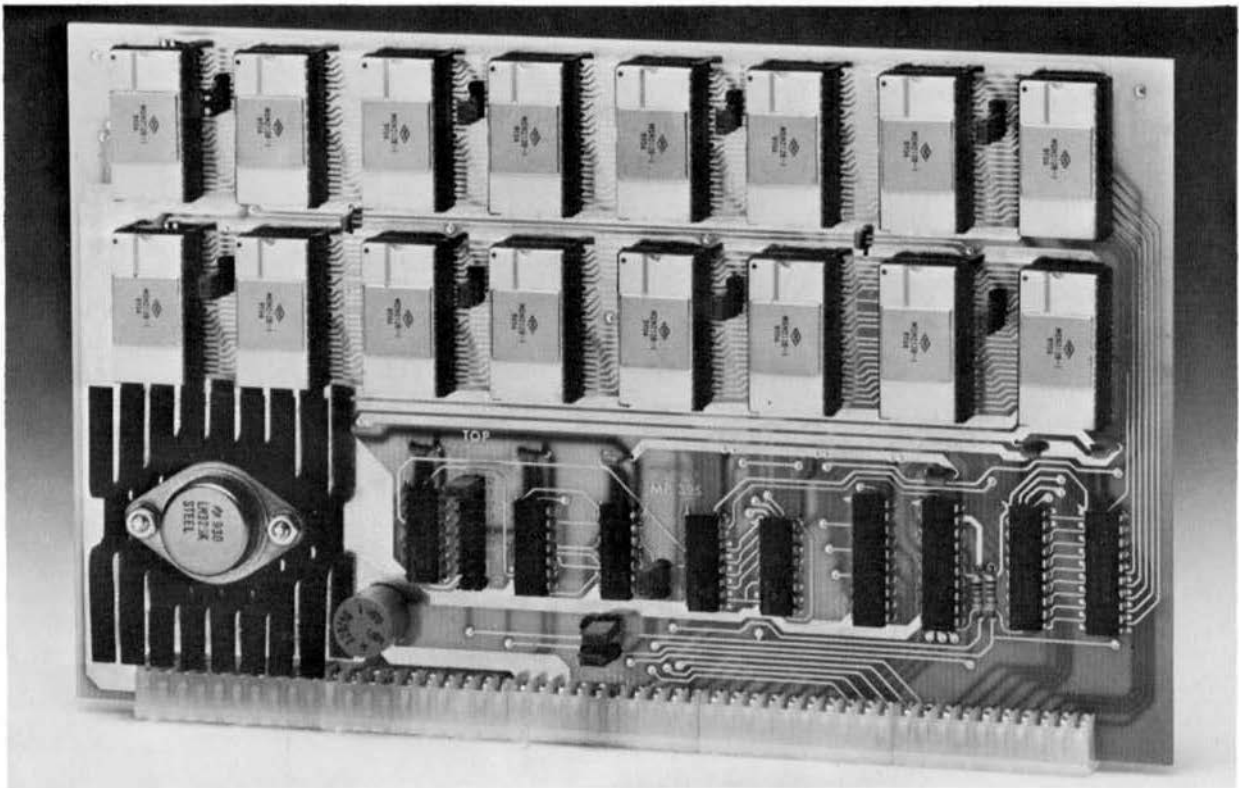
PERIPHERALS— The wide range of peripheral hardware that is supported by the 6809 includes: dot matrix printers (both 80 and 132 column), IBM Electronic 50 typewriter, daisy wheel printers, 5-inch floppy disk system, 8-inch floppy disk systems and a 16 megabyte hard disk.

SOFTWARE— The amount of software support available for the 6809 is incredible when you consider that it was first introduced in June, 1979. In addition to the FLEX9 operating system, we have a Text Editor, Mnemonic Assembler, Debug, Sort-Merge, BASIC, Extended BASIC, MultiUser BASIC, FORTRAN, PASCAL and PILOT.

69/K Computer Kit with 8K bytes of memory	\$ 495.00
69/A Assembled Computer with 8K bytes of memory	\$ 595.00
69/56 Assembled Computer with 56K bytes of memory	\$1,495.00



SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241



UNIVERSAL static memory card

- ★ 32K bytes-ROM, RAM, EPROM or a combination
- ★ SS-50 A&C compatible with 16 and 20 bit address decoding
- ★ Compatible with all SWTPC 6800 and 6809 computers

This is the most versatile memory card you can buy. Our S-32 may be populated with up to 32K of static RAM, EPROM, or ROM, or any 4K block combination of these that you may desire. Any 5-volt 2716 pinout compatible memory may be used in this card. Any 4K block of the memory may be jumper block programmed for RAM or ROM use. This feature makes this the ideal memory for those process control applications that require a mixture of ROM and RAM memory. The board is fully compatible with all SWTPC 6800 and 6809 computers.

The power requirement for the board is only 1.75 amps at 5.0 volts with a full 32K of RAM installed.

S-32 Circuit card assembled

—less memory IC's (uses up to 16). \$99.50

2716 Type EPROM for above \$50.00 ea.

16K (2K x 8) Static RAM for above
(4016 or 2128). \$50.00 ea.



SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

long as the memory allocated for the program being loaded does not coincide with whatever memory location at which you choose to org TRAP. I thought about placing it in the FLEX transient command memory space, but since I do considerable writing and modification of transient commands which I often need to debug, I decided to place it in high memory and have found that in general I have no problems with target programs writing over it. Once the target program is loaded, TRAP has done its job, therefore, there is no problem with TRAP being destroyed by programs which do available memory tests.

The first operation performed by TRAP is obtaining the target program file information. If no file extension is specified the default is a ".BIN" extension. If an error is made in the target program's file specification, a "FILE SPEC ERROR" message is output and the program returns control to FLEX. If no errors in the file specification are encountered then the file is opened for reading using the FLEX file management system. Any errors encountered when the file is opened will result in the appropriate FLEX error message with control being returned to FLEX.

The program uses the FLEX LOAD subroutine to load the target program into memory. The LOAD subroutine has the capability to load programs with a fixed address offset.

Since no offset is desired, the offset location must be cleared. Subroutine \$755D is an undocumented FLEX subroutine which will accomplish this by simply doing a JSR to this location. Also the space compression flag which resides at location \$3B in the file control block must be set to the uncompressed file state (\$FF) for loading binary files. The Advanced Programmer's Guide writes up for the LOAD subroutine does not do a very good job in making it clear that this must be done to obtain proper loading. Because of this I wasted several hours trying to understand why my target programs would not load correctly. If you plan on using the LOAD subroutine for anything you need to make sure this flag is set correctly. The LOAD subroutine supplies two other pieces of information. That is a link address and a flag indicating whether a linkage address exists. The link address specifies the starting address for the target program and is utilized by TRAP as the address for the program counter placed on the stack. If no link address exists a "NO LINK ADDRESS" message is output and control returns to FLEX. If a link address is found it is incremented by one and placed on the stack as the address of the program counter. The stack is pushed an appropriate number of times to satisfy the number of items on the stack required for an "RTI" and control is transferred to the monitor. The user is now free to make any changes to the target program object code he wishes. Typing a "G" will cause the monitor to initiate a "RTI" which will restore the state of the processor to that which existed before the pseudo software interrupt occurred. Since the target program's start address was put on the stack as the program counter, control will be transferred to the target program. If the target program contained file specification data the input buffer pointers in FLEX will point to the proper data and it will be available to the target program as though the data was entered as part of a normal input command line.

EXAMPLE

As an example of how the TRAP program is used let's assume that we have a program called TEST which starts at location \$0100 and requires file input specification data to execute. If TRAP resides on the system drive as a CMD file and TEST resides on the work disk as a CMD file, simply type:

```
TRAP,TEST,CMD,DUMMYPIL
```

The program would be loaded and control transferred to the monitor which would respond with the standard software interrupt monitor output. Data for the program counter would contain \$0100, the start of the program. Program modifications such as program patches and software interrupts can now be made and execution begun by typing "G". The input line buffer will point to the first character of DUMMYPIL and its file specification information will be available to the target program.

MODIFICATIONS TO OPERATE WITH OTHER MONITORS

TRAP was written to run on a system which has MIXBUG for a monitor program. It will also run properly as written, on systems which use \$BARTLEBY for a monitor. If you have a system which uses \$MIBUG it will be necessary to make the modifications indicated in the listing to make TRAP work properly. This will set up the breakpoint jump address such that the pseudo software interrupt will operate properly.

TSC ASSEMBLER PAGE 1 18/3/79 12.3

TSC ASSEMBLER VER 1.12
INPUT FILE IS TRAP.TXT 8/13/79 ON DISK SCRATCH LAJ.B05A 8/24/79

```

1      NPM      TRAP
2      OPT      PAG
3
4      * WRITTEN BY L. A. JOHNSON
5      * PARISON, IOWA
6
7      * THIS PROGRAM IS DESIGNED TO PROVIDE
8      * CONTROL TO THE USER AFTER A PROGRAM
9      * HAS BEEN LOADED INTO MEMORY BUT
10     * BEFORE EXECUTION BEGINS. ALL DATA
11     * ENTERED INTO THE INPUT LINE BUFFER IS
12     * RETAINED.
13     * THIS ALLOWS THE USER TO LOAD A DOS
14     * PROGRAM WITH FILE SPECIFICATIONS AND
15     * MAKE PATCHES WITH THE SYSTEM MONITOR BEFORE EXECUTION.
16     * OF THE TARGET PROGRAM BEGINS.
17     * THIS PROGRAM IS WRITTEN WITH THE SMP 6800 SYSTEM
18     * AND THE MIXBUG MONITOR.

```

26

```

19     6000      OPB      #6000
20     7127      GETFIL   EQU      #7127
21     712D      SETEXT   EQU      #712D
22     712A      LOAD     EQU      #712A
23     7740      FCB      EQU      #7740
24     7806      FMS      EQU      #7806
25     789E      LINK     EQU      #789E
26     789F      SRTADR   EQU      #789F
27     7118      PSTAND   EQU      #7118
28     713C      RPTERR   EQU      #713C
29     7803      FMSCLS   EQU      #7803
30     7103      WRTMPS   EQU      #7103
31     7950      CLROFF   EQU      #7950
32     6FFA      SFE      EQU      #6FFA
33     8001      VERSION   EQU      #8001
34     8000 20 01  START   BRR      #8000
35     6802 01      PCB      VERSION
36
37
38     8003 CE 77 40  BRRIN  LDX      #8003
39     8004 80 71 27  JSR      #8004
40     8005 25 3E      BCS      #8005
41     6000 CE 77 40  LDX      #6000
42     600E 4F      CLR      #600E
43     600F 8D 71 2D  JSR      #600F
44     6012 CE 77 40  LDX      #6012
45     6015 86 01      LDA      #6015
46     6017 A7 00      STA      #6017
47     6019 8D 70 06  JSR      #6019
48     601C 26 33      BNE      #601C
49     601E 8D 70 30  JSR      #601E
50     6021 CE 77 40  LDX      #6021
51     6024 86 FF      LDA      #6024
52     6026 A7 30 1      STA      #6026
53     6028 8D 71 2A  JSR      #6028
54     602B 86 70 9E  LDA      #602B
55     602E 27 26      BEQ      #602E
56
57     *SET UP PROCESSOR STACK AS THOUGH SOFTWARE
58     *INTERRUPT OCCURRED. THE START ADDR FOR THE
59     *PROGRAM IS STORED IN THE PROGRAM COUNTER LOCATION
60     *SO THAT A RTI WILL INITIATE EXECUTION OF THE
61     *TARGET PROGRAM.
62
63     6030 FE 70 3F      LDX      #6030
64     6033 08      INX      #6033
65     6034 FF 70 3F      STX      #6034
66     6037 86 70 30  LDA      #6037
67     6039 26      PSW      #6039
68     603B 86 70 3F  LDA      #603B
69     603E 36      PSW      #603E
70     603F 36      PSW      #603F
71     6040 36      PSW      #6040
72     6041 36      PSW      #6041
73     6042 36      PSW      #6042
74     6043 36      PSW      #6043
75     6044 FE FF FA  LDX      #6044
76     6047 62 00      JMP      #6047
77
78     6049 CE 60 62  FILEERR LDX      #6049
79     604C 8D 71 18  JSR      #604C
80     604F 20 00      BRR      #604F
81     6051 8D 71 3C  FMSERR  JSR      #6051
82     6054 20 06      BRR      #6054
83     6056 CE 60 72  LNKEPR  LDX      #6056
84     6059 8D 71 18  JSR      #6059
85     605C 8D 70 03  ERRADR  JSR      #605C
86     605F 7E 71 03  JMP      #605F
87
88     6062 46      GETERR   FCC      #6062
89     6071 4E      FCB      #6071
90     6072 4E      LNKMSG   FCC      #6072
91     6081 04      FCB      #6081
92
93     *-----
94     * INSERT NEXT LINES IF USING TRAP ON A SYSTEM
95     * WITH $MIBUG FOR A MONITOR
96
97     *      ORG 80012
98     *      FDB 0E123
99
100    *-----
101    END START

```

NO ERROR(S) DETECTED

TAPE LABELING

'6800' Micro Journal
3018 Hamill Rd.
PO Box 849
Mixon, TN 37343

Dear Sir:

For the past several months, I have been using the JPC high speed cassette interface and their file manager CFM/3 with considerable satisfaction. I have recently added the TSC full featured BASIC interpreter to my system with which I had earlier used the TSC Micro Plus and the Computerware BASIC interpreters. As you might suspect, I now have a goodly number of cassette-stored programs and their labeling and grouping has been a housekeeping chore of some substance. The attached listing and description give in creation on a labeling program I have been using that may be of some interest to your readers.

I enjoy your magazine! Keep up the good work!

Sincerely,

Ches Looney
Ches Looney

'68' Micro Journal

This program is useful for printing title cards for insertion into computer program or audio cassette boxes. The test run shows a cassette card near the end of the run. Two rows of hyphens set off the two-line cassette title and indicate the foldlines lines so that the box title will show at the edge while the individual program or tape titles show through the transparent sides of the cassette box. The printer used (Heathkit H14) prints 6 lines per inch and permits software selection of one of three print pitches. The program selects 10 characters/inch for the cassette titles and 16.5 characters/inch for the program titles.

The program is written for use with Computervare's cassette version of BASIC. The HOME command in line 10 clears the CRT screen; lines 15 and 20 disable the line length control (LINE=0), dimension for title lengths of 34 characters and a total of 26 titles, and print a program title of "INDEX". Line 35 indicates the use of the exclamation point (!) to signify the end of title entry. Line 40 calls the subroutine at line 400 to assemble the individual titles and the two-line box title. The expression, B=I-1, in line 48 is to adjust the variable B to the number of titles entered. Computervare's interpreter returns the value of the variable in a FOR-NEXT loop as one greater than the full count. The expressions starting with "PRINT CHR\$(27)" in lines 45, 55 and 65 control the print pitch variable in the printer. The two commands are "ESC u (" for the 16.5 character/inch and "ESC u Control-A" for 10 characters/inch. ESC is the "ESCAPE" command or character and is identified by the CHR\$ command as "27"; similarly, Control-A is identified as CHR\$(1).

The two lines of hyphens are printed as a subroutine in line 500 and are called by lines 50 and 70. If there are less than 14 titles, only one side of the title card is printed. This is controlled by the "IF B<14" commands in lines 120 and 150 and by the "IF A=C<B" command in line 170. The INPUT command in line 190 is used only to hold the finished title card on the CRT monitor and to avoid printing the BASIC prompt message at the end of the run until the printer has been tended to and the card removed. Lines 210 through 230 permit corrections to be easily made.

The subroutine starting at line 400 asks for up to 26 titles, and advises the user when the second column or right hand side of the title card is about to be started. The advisory message is in line 405 and is called by the "IF A=14" command. Line 410 stops the program or tape title entry with the "IF A\$(A)..." command. Line 420 adjusts the length of A\$(A) to be 29 characters or less, although the program has permitted up to 34 to be stored. This variable can be used to adjust output to other computer line pitches. Lines 430 through 470 request the cassette box two-line title and center it. The expressions starting with "PRINT CHR\$(27)" in lines 45, 55 and 65 control the print pitch variable in the printer. The two commands are "ESC u (" for the 16.5 character/inch and "ESC u Control-A" for 10 characters/inch. ESC is the "ESCAPE" command or character and is identified by the CHR\$ command as "27"; similarly, Control-A is identified as CHR\$(1).

The two lines of hyphens are printed as a subroutine in line 500 and are called by lines 50 and 70. If there are less than 14 titles, only one side of the title card is printed. This is

controlled by the "IF B<14" commands in lines 120 and 150 and by the "IF A=C<B" command in line 170. The INPUT command in line 190 is used only to hold the finished title card on the CRT monitor and to avoid printing the BASIC prompt message at the end of the run until the printer has been tended to and the card removed. Lines 210 through 230 permit corrections to be easily made.

The subroutine starting at line 400 asks for up to 26 titles, and advises the user when the second column or right hand side of the title card is about to be started. The advisory message is in line 405 and is called by the "IF A=14" command. Line 410 stops the program or tape title entry with the "IF A\$(A)..." command. Line 420 adjusts the length of A\$(A) to be 29 characters or less, although the program has permitted up to 34 to be stored. This variable can be used to adjust output to other computer line pitches. Lines 430 through 470 request the cassette box two-line title and center it.

```
0010 HOME:PRINT CASSETTE INDEX: 6/79 ch1
0015 LINE= 0:GOTO A$(26):A(26)
0020 HOME:STRING$=34:PRINT TAB(10);"INDEX":PRINT
0030 PRINT "ENTER TITLES IN SEQUENCE; THE FOURTEENTH WILL"
0035 PRINT "START THE SECOND COLUMN. USE '!' TO STOP ENTRIES."
0040 GOSUB 400:B=A-1:INPUT "TYPE 'RETURN' TO PRINT":G6
0045 PRINT CHR$(27);"u"
0050 GOSUB 500
0055 PRINT CHR$(27);"u":CHR$(1)
0060 PRINT B:PRINT C$
0065 PRINT CHR$(27);"u"
0070 GOSUB 500
0100 PRINT TAB(4);"==== TITLE =====:IF B<14 PRINT:GOTO 120
0110 PRINT TAB(30);"!":TAB(34);"==== TITLE ====="
0120 FOR X=1 TO 29:PRINT " ":NEXT X:IF B<14 PRINT:GOTO 140
0130 PRINT " ":FOR X=1 TO 30:PRINT " ":NEXT X:PRINT
0140 IF B<13 C=13
0150 IF B<14 C=8
0160 FOR A=1 TO C:PRINT A$(A):
0170 IF A=C<B PRINT TAB(30);"!":TAB(32);A$(A+C):
0180 PRINT:PRINT A
0190 INPUT "PRINT CARD AGAIN?":G6
0200 IF LEFT$(G6,1)="" GOTO 45
0210 INPUT "IF CORRECTION NEEDED, ENTER TITLE NUMBER":Z$
0220 IF Z="" END
0225 IF A$(Z)>57 END
0227 IF VAL(Z)>26 END
0230 LET A=VAL(Z):PRINT A$(A):INPUT "NEW TITLE":A$(A):GOTO 190
0400 FOR A=1 TO 26
0405 IF A=14 PRINT "NEXT TITLE IS TOP OF SECOND COLUMN."
0410 PRINT "TITLE "A:INPUT A$(A):IF A$(A)="" GOTO 430
0420 LET A$(A)=LEFT$(A$(A),29):NEXT A
0430 INPUT "TYPE CASSETTE TITLE FIRST LINE":G6
0440 INPUT "TYPE CASSETTE TITLE SECOND LINE":C$
0450 FOR X=1 TO (34-LEN(B)):Z$="" :G6:NEXT X
0460 FOR X=1 TO (34-LEN(C)):Z$="" :C$:NEXT X
0470 RETURN
0500 FOR X=1 TO 60:PRINT " ":NEXT X:PRINT:RETURN
```

BASIC
RUN

INDEX

```
ENTER TITLES IN SEQUENCE; THE FOURTEENTH WILL
START THE SECOND COLUMN. USE '!' TO STOP ENTRIES.
TITLE 1 ? 3 - OREGON TRAIL
TITLE 2 ? 87 - DISCOVERY
TITLE 3 ? 185 - UNBLOTT
TITLE 4 ? 210 - UNBLOTT2
TITLE 5 ? 237 - CASSETTE INDEX
TITLE 6 ?
TITLE 7 ?
TITLE 8 ?
TITLE 9 ?
TITLE 10 ?
TITLE 11 ?
TITLE 12 ?
TITLE 13 ?
NEXT TITLE IS TOP OF SECOND COLUMN.
TITLE 14 ? 3 - SORCERY
TITLE 15 ? 75 - TWO DIAMONDS
TITLE 16 ? 105 - FUN
TITLE 17 ? 152 - RISK CHART
TITLE 18 ? 167 - QUEST
TITLE 19 ? !
TYPE CASSETTE TITLE FIRST LINE? ROM BASIC
TYPE CASSETTE TITLE SECOND LINE? GAMES - U
TYPE 'RETURN' TO PRINT?
```

ROM BASIC GAMES - U

==== TITLE =====	==== TITLE =====
3 - OREGON TRAIL	3 - SORCERY
87 - DISCOVERY	75 - TWO DIAMONDS
185 - UNBLOTT	105 - FUN
210 - UNBLOTT2	152 - RISK CHART
237 - CASSETTE INDEX	167 - QUEST

Letters—New Products—Etc.

***** HELP *****

Dear Sir,

Needed: Information on where to get or how to add expanded graphics capability to a SWTPC 6800 System with CT-61 (Would like graphics similar to PET). Also: Information on adding Speech Synthesis to SWTPC 6800 System.

Sincerely,
A.T. Magill Chm. Sci. Dept.
Streator Township H.S.
Streator Ill. 61364

Help,

I work extensively with the SWTPC AC-30 using it for all manner of I/O. A recent program of mine makes and incorporates an effort to store some data from tables. I am running into miserable problems that haven't occurred before when trying to I/O records that incorporate more than a few variables from or to the tables. Can anyone offer their experience?

Thank You,
J.O'Loughlin
908 Salem Dr.
Huron Ohio 44839

*** HELP FOLLOW-UP ***

Dear Mr. Williams,

I would like to thank you for printing my letters in your HELP Column. I have received many responses from fellow SWTPC 6800 users who have been kind enough to give me solutions to the problems I was encountering. Clearly they have been down the same road before and their advice has made running my system more effective, less troublesome, and simpler. I have received software modifications that allows the port command in SWTPC Cassette Basic V.2 to be useable as well as Printer Routines to be used with TSC Text Editor and Processor. I have installed a MP-S card on port 2 of my system and loaded it with a L.E.D. and resistor to check for transmission of data at that port. It was great when that L.E.D. flickered when I used the port commands in Basic or the Printer Routines provided by a fellow user for the TSC Editor and Processor and regained control at the CT-64. I am waiting for my Worldwide Electronics BCD to

ASCII Serial I/O Printer to arrive and I believe I am better prepared to use this device thanks to the response by readers of '68' Micro Journal. Regarding the generation of spaces on the CT-64, this was found to be a key switch on the space bar sticking in the 'ON' position. I am still working with the cassette in spite of the greater availability of software on disk. By the way I've dumped the TSC Editor and Processor to one cassette. To anybody else who wishes to do this rather than go through the LOAD/RELOCATE- this and that Routine use the following memory contents:

P MA048-MA049 (2000)
P M0097-M009D
P M00B3-M00B6
P M0200-M15D0
P M2000-M330D
Includes Space For Printer Routine

Now I can load TSC Editor and Processor in one third the time.

I've enclosed a small Routine (Basic) for adding up a grocery bill or summing cashed checks. Also enclosed is a review of AAA Chicago Cassette K Basic.

Will Sum Up To 100 Items

```
060 PRINT "TO ADD UP ENTER 0"  
070 S=0  
080 DIM A(100)  
090 FOR N=1 TO 100  
100 PRINT "ENTER ITEM COST"  
110 INPUT A(N)  
120 IF A(N)=0 THEN GOTO 150  
130 S=S+A(N)  
140 NEXT N  
150 PRINT "THE NUMBER OF  
ENTRIES ARE:";N-1  
160 PRINT "THE SUM OF  
THOSE ENTRIES IS:";S  
170 END
```

P.S. Other users can modify to Include Tax Rates etc.

Thank You,
Jeffrey M Craig
Apt 912-3001 S.King Dr.
Chicago, Ill 60616

*** REVIEW ***

AAA Chicago Computer Center sells a Cassette Basic K that has all of the features of the SWTPC Cassette Version plus Line Renumber, Delete Lines, and Print Using. If you have used SWTPC 8K Cassette Basic you will have no problem in using AAA's K Basic. The statements, commands, and format are the same. Once the K tape has been loaded (20 minutes) and brought up one can load programs developed with SWTPC 8K V.2 Basic and then modify them with such commands as RENUM and DELETE M,N. It is really nice to see every line spaced by 10! Another advantage of K Basic is that line length is increased to 74 Characters. 9K of memory is required for this Basic as well as a 32K system (there is a routine in the 7000's). Again this version like SWTPC's 8K V.2 Cassette Basic has port commands that are inoperable. A simple change of two memory locations will allow one to backspace with a control H. However when this modification was implemented, programs with heavy math content were unloadable. The patch command was tried to get (MIKBUG) but the cursor was left hanging. Programs developed with SWTPC 8K V.2 Cassette Basic were run on K Basic and they ran OK.

This Cassette Version of Basic is \$55. There is a Disk Version (not received here and to which some of the above statements may not apply concerning Patch and control H loading problems) at \$50.

System: SWTPC 6800, 32K RAM, MP-A, MIKBUG, MP-C on port 1, AC-30, CT-64, Craig Cassette Recorder.

Jeff Craig
Apt 912-3001 S. King
Chicago, Ill 60616

*** CLASSIFIED ***

SWTPC 6800,SWTBUG,STANDARD I/O with manuals \$450 or make offer Call Larry Clark (615)842-7655

SWTPC Memory Boards, Low Power Chips-Fully T.l. Socketed, only 5 left-\$50 each: Ted Wolff, 579 W. 215th St., NY NY 10034

New Gimix Chasis Video SSB 16K Memory MP1 B51 etc. \$1700 also B&K Oscopce \$400, M.J. Labombard, 1327 Scott Dr, National City, Ca. 92050

SWTPC CT-82 terminal with graphics,\$700 postpaid Phil Hughes, Box 2847, Olympia, WA 98507 Days (206)357 4415 Evenings (206)352-9637

SWTPC MP 09 processor board \$150. FLEX 9.0, EDIT, ASMB for MF 68, \$70, Phil Hughes, Box 2847, Olympia, WA 98507 (206)357 4415

WEST COAST COMPUTER FAIR
March 14,15,16, 1980

The annual 1980 West Coast Computer Fair is to be held at the San Francisco Civic Center this March 14th, 15th, 16th. There will not be as many S50 bus 68XX dealers as in previous years. However we (68 Micro Journal) will be there, in and out of the GIMIX booth (104), and hope to meet a lot of our readers and others we have not had the pleasure of shaking hands with. Please look me up, I will be wearing a hat that has imprinted 68 MICRO JOURNAL on the front. If I happen to be out of the booth leave a message or look for the red white and blue cap. I look forward to seeing many of you there.

While on the subject of the GIMIX booth I hope that all of you will get a chance to drop by and also meet Richard Don (Mr. GIMIX), Ken Kaplan (president of MICROWARE) and possibly Terry Ritter (Project Manager for the Motorola 6809 development team). Microware developed the OS"IX" System and BASIC09. I understand that GIMIX will have running the new OS"IX" Operating System and BASIC09 language system for the 6809, with the new GIMIX 6809 MPU board. This software was developed by Terry and Ken for Motorola and will be available through Motorola distribution channels, in the future. Our initial information indicates that the above will include comprehensive management of memory, MPU time and all I/O communications.

The system runs in two basic packages (ROM). Level one (1) is a single user single memory map version, and is a compatible subset of level two (2). Level two (2) is a multiprogramming, real time operating system that supports full timesharing on systems equipped with extended memory mapping and an interrupt-driven disk system. Both versions are system independent and will run on most available 6809 systems.

The OS"IX" I/O is similar to UNIX" (Bell Telephone Labs) with many enhancements for modern hardware environments. Level two supports

inter-process communications and a hierarchical directory system. ROM 'kernels' are position independent, running anywhere in available memory.

Also demonstrating at the GIMIX booth will be Dave Allen of Control Systems Inc., Microsystems Division, who will be demonstrating the current 6809 version of UCSD PASCAL", running on a SWTPC 6809 S50 bus computer system. It is our understanding that this is one of the most complete version of UCSD PASCAL" running on any micro. While on the subject of GIMIX, rumor has it that their new 512X512 direct bit graphics memory board is soon to be available. It is a two board set and uses a modified GIMIX 32K static memory board as one board. We have used the GIMIX Super Video Board for over 6 months now and if the new video board works as well and glitch-free as the one we have been using, it should do well.

If you make the West Coast Computer Fair be sure and catch the talk to be given by Terry Ritter of Motorola. The subject is "Modular and Structured Programming on Small Systems Including 6809 Assembly Language". This should be a must for all programmers of the 6809. A hard combination to beat; as Terry is the developer of the 6809 and the 6809 is by far the best micro going today!

We hope to have a complete review of the software and ROM packages as soon as they become available to us. See you at the Fair.

DMW

TO: 68 Micro Journal
3018 Hamill Rd
Hixson, TN 37342

Attn: Letters to the Editor

Dear 68XX Users,

LET'S ALL GET ON THE S550 BUS

A funny thing happened to me while I was selling some chips. An S100 memory board manufacturer phoned to buy some 4044s from me. After he placed the order he asked me how we were doing selling memory cards for the S550 bus, and why we had decided to make such an odd bus, since we were the only ones selling boards for that bus. He was surprised to learn that SWTP was on the same bus, as well as so many other manufacturers. I later looked at the ads of other S550 makers and saw that they seemed to be keeping a good secret of what bus they were on. Now, for the readers of the 68 Micro Journal they already know who is making what, but for readers of the general computer magazines, a lot of them do not know whose boards are compatible with other makes.

SWTP started the S550 and all the other S550 manufacturers that I am aware of started by making boards for the market that they originated. We have a good thing going for us by being able to brag about our compatibility. Why, then, do some of my fellow manufacturers want to hide the fact of which bus they are on? Or could it be that they are assuming that all the users in the general marketplace know that the S550 is the compatible bus and who is on it.

GIMIX IS PROUD OF THE FACT THAT WE ARE ON THE S550. We advertise that we are compatible with SWTP.

We view the enemy in the marketplace as the non SS50 bus machines. When a user buys one of the enemy machines, we have most likely lost him for all time as a customer. BUT--When someone buys an SS50, we will in all likelihood have him for a future customer for one of our boards, regardless of whose mainframe he originally buys. And, the same holds true for our fellow SS50 bus manufacturers.

So let all of us, readers, manufacturers, software houses, and anyone else interested in promoting the SS50 bus, proudly tell the world out there that we're all riding the best bus there is.

Richard Don
GIMIX INC.
1337 W. 37th Place
Chicago, Ill 60609

168 Micro Journal
3018 Hamill Road
Hixson, Tenn.

Gentlemen:

REALLY: I would have thought that a letter which comments about an error you published and makes the statement that it is "devastating to K6800 systems" (Nov./Dec/ 1979, p. 23. Mr. Davis) would have attracted your attention.

It would not have taken long to see that Mr. Davis' problem (or confusion as it were) is self-inflicted. He is apparently unaware that a T8X instruction automatically adds one to the SP value prior to placing it in the XH. Consequently, a LDX 0,X works fine for retrieving the return address of a JSR. Mr. Davis' problem, for which he corrected nicely, is his PSHA prior to doing the T8X. This decremented the SP by one more causing him to compensate by doing a LDX 1,X.

But you are forgiven your oversight as indicated by the enclosed subscription renewal. Keep up the good work. I really enjoy your publication.

Sincerely,
Mike Lyddane
Mike Lyddane
San Jose, Calif.

December 15, 1979

168 Micro Journal
3018 Hamill Road
PO Box 1448
Hixson, Tennessee 37343

Personnel:

I would like to submit a one wire modification note for the Micro Works PSB-08 2703 EPROM board to utilize the 16 Pin within the 32 EPROM address.

The PSB-08 is a 68K 2703 EPROM and a 16 Pin 2102 68K Board. Both the EPROM and P801 could be addressed to any 68 block of memory. For example: 8000h, 8200h, 8400h, 8600h.

The EPROM section of the board will fit nicely within a 68K block. But for the P801 on the other hand, only 16 is used, 76 wanted.

For those people who would like to have the extra 16 bit can not sacrifice an 8K block for it, the P801 would just have to be disabled. Well, if you have one EPROM socket not used, and one wire, the 16 could be remenable within the EPROM address block.

Here is how:

Simply make a jumper from U27 pin 5 to whichever free EPROM socket pin 20. Disable the P801. And you have it, one extra 16.

This is how mine is setup, use it as an example:

EPROM addressed at 8E000.
Jumper from U27 to U5.

Pin	Comment	16 block
1	Used for my monitor (REBUS)	8E000 to 8E0FF
2	Also used for monitor	8E100 to 8E1FF
3	Not used *	8E200 to 8E2FF
4	Not used *	8E300 to 8E3FF
5	16 Pin *	8E400 to 8E4FF
6	Not used *	8E500 to 8E5FF
7	EPROM burner address	8E600 to 8E6FF
8	CPU vectors, memory test, and system startup ID messages	8E700 to 8E7FF

* Available for the 16 Pin.
** 16K at EPROM location.

I would like to take this time if I may, to thank you for a great journal, and MESA for the technology.

Robert T. Leon
Robert T. Leon
705 North Linden Drive
Beverly Hills, California 90210

REPLY TO:
ATTN OF: CAPT F G TOWER
271 10TH ST
HOGDEN, UTAH 84404

9 JAN 1980

SUBJECT: DIGITAL RESEARCH: COMPUTERS

TO: 68 MICRO JOURNAL

1. SPECIAL THANKS SHOULD GO TO JIM TANNER AND DIGITAL RESEARCH: COMPUTERS (OF TEXAS) FOR THEIR SINCERE CUSTOMER RELATIONS POLICY. I URGE ALL TO CONSIDER DOING BUSINESS WITH THIS FINE FIRM.

2. I ORDERED THEIR SWAP BY BOARD AND WAS GIVEN A TWO WEEK DELIVERY. AFTER FIVE WEEKS I CALLED TO CANCEL MY ORDER. THEY APOLGIZED AND SAID THEY WOULD REFUND MY MONEY. TWO DAYS LATER I RECEIVED MY REFUND PLUS THREE DOLLARS FOR MY PHONE CALL AND A GENEROUS DISCOUNT ON MY NEXT ORDER.

3. NOW THAT IS THE TYPE OF COMPANY THAT I CAN DO BUSINESS WITH. THEY PROMISED ARRIVE AND BEYOND THE SIMPLE REFUND. THEY MADE A CONSCIENTIOUS EFFORT TO APOLGIZE WHICH WAS UNEXPECTED BUT SINCERELY APPRECIATED.

Francis G Tower

FRANCIS G TOWER
CAPTAIN, USAF

Ed's Note:

We made a call to Jim Tanner of Digital Research: Computers and were told that during that particular time the memory devices were on backorder (by factory) and did cause a period of delay. They are now current and are able to ship within two weeks. We would like to note that we have received many letters and calls, from readers, telling us how much they like the memory board. Because of our previous 'Lab Review' of the Digital Research: Computers 16K memory kit, and the high rating given, the above is published for reader benefit.

DMW

JPC PRODUCTS COMPANY

P.O. Box 5615 • Albuquerque, New Mexico 87185 • Phone (505) 294-4623

MEMO-LETTER

TO: GLENN HITT
Route #1 Box 309-A
Reno, NV 89305

DATE: 12 Dec 79
BY: Sam Rader

Glenn, I read into expansion of Page A and found a nice Schick - Enchanted is a copy of a letter to Dr. Williams which explains the procedure. How about it and if it works well!
After you install the new C810 remember to change your dip switch on the A2-board (Switch no 6 off)

Sam Rader

Mr. Don Williams
'68' Micro Journal
Nixon, TN 37343

SCRATCH PAD RAM EXPANSION

December 12, 1979

Dear Don,

One of our customers called and asked if it was possible to expand the memory space in block \$A000 without installing a 4K board. He had our CFM/3 Cassette Operating System EPROM Version 2.0A and had dedicated one of his 4K boards to accommodate that version's scratch pad memory (\$A080-\$A0FF). He was looking for an easier (cheaper) way to get the \$80 Bytes.

IC-3 on the MP-A2 board is a Motorola MCM6810 which has \$80 BYTES of RAM so the expansion of this \$80 Bytes of memory was the logical choice. The job turned out to be very simple. The decoding of the 6810 leaves a number of chip enable lines permanently grounded so that it answers to a number of address blocks (rolls over every \$80 bytes). By using one of these enable lines it was possible to expand the scratch pad RAM by just "piggy-backing" another 6810 and using one of the spare enable pins to select the proper chip.

The following is the procedure that was used:

- (1) Get another 6810 (cost about \$10.00)
- (2) "Wink Out" pin #11 on both 6810's
- (3) "Piggy-Back" solder the remaining pins together
- (4) Attach two 7" wires to the #11 pins (light gauge wire)
- (5) Drill a hole in the MP-A2 board between IC-3 & IC-2 (this is optional but made it look neater)
Place the hole just to the right of IC-3 pin #11 and be careful of the trace behind the board!!!
- (6) Feed the two wires from the 6810's through the hole and attach one of them to the A7 (Address line #7) pin on the bottom Molex connector.
- (7) Put a jumper wire from the same A7 pin on the Molex to IC-9 pin #13 (7404).
- (8) Put the second 6810 jumper wire to pin #12 on IC-9.

IC-9 is a HEX Inverter and one of the gates is not used on the A2 board. By using this gate on A7 it saves A7 and allowed it to select one 6810 as \$A000-\$A07F (just like the original 6810 decodes worked) and the other as \$A080-\$A0FF which meets the requirements of JPC's CFM/3 Version 2.0A. This would also help any one that needs a few extra bytes out of the reach of other programs.

This was all that was required. Any one doing this mod should be VERY careful on all soldering and STATIC ELECTRICITY!!!!

WB9FRV

Ken Van Andel
1803 California Avenue
Aurora, Illinois 60506, U.S.A.
1 November 1979

68' Micro Journal
3018 Hamill Road
P.O. Box 849
Nixon, Tennessee, 37343

Gentlemen,

Since your staff is busy, three short comments to you!

- 1-Yes, many of us in the area who missed the first few issues would be interested in buying a reprinted volume of the first several issues to complete the collection.
- 2-Reference the contest---much confusion here about the BASIC division. Does this mean programs written in BASIC or software patches to existing BASIC interpreters? Perhaps you could clarify in next issue.
- 3-Many of us little guys are up and running with 6800 and have no intention of waiting for--or time to reprogram for 6809 systems. Therefore, don't get so caught up with 6809 news and programs that you neglect a large base of 6800-forever hackers!

Keep up the good work---all of you! Having 6800 only type ads is worth price of subscription alone!

Sincerely yours,

Ken Van Andel

Amateur Station W6KMI
Julian "Jerry" Faas
4113 East Tyler Avenue
Fresno, California 93702 1 Jan. 1980

Gentlemen,

I have a STP 6800 computer that I made from a kit with sockets for all IC's, it consist of STP/1 kit, with a AC-30, CT-64 terminal with CT-VM 9 inch monitor.

I have a SEALS 68K5C memory board giving me a total of 20K of memory.

I have the MF-68 Floppy Disk with the Flex program.

I have a SWTBWG in the MP-A board and use it with a MP-C and a MP-LA board for RTTY with a program by Rich Kotz K9AR also have his microtime board that give me the time via RTTY.

I am thinking about getting rid of this system as at the ago of 70 the programing has been a little over my head and there is no local help.

Would accept any reasonable offer! I am using the system with my ham station on RTTY.

73
John P. Tucker

JOHN P. TUCKER
Post Office Box 2898
Laredo, Texas 78040
November 2, 1979

Don Williams
'68' Micro Journal
Post Office Box 848
Nixon TN 37343

Dear Sir,

With the belief that a good job well done deserves to be recognized and publicized to further our hobby, I am sending a copy of this letter to the publishers of KiloBaud, Byte, and '68' Micro Journal. If at all possible, I would like for each of them to give some space to the following note:

I was one of the very first purchasers of the TSC BASIC FOR 6800 from Technical Systems Consultants of West Lafayette, Indiana. NEVER HAVE I RECEIVED SUCH CO-OPERATION AND HELP FROM A SUPPLIER AS I HAVE FROM TECHNICAL SYSTEMS CONSULTANTS! The disk version I received, one of the first sent, had a byte in error causing a malfunction of such functions as MIDS, LEPTS, etc. An overly long letter to them netted a prompt reply -- my disk version needed one byte changed which was a task even I could accomplish. Then one of my more technical BASIC programs, under certain conditions, would cause the BASIC to error out falsely. This time I tried a more business-like approach and sent along data to support my questions (including a cassette copy of the program in question). Once again TSC spent valuable time in researching the program and sent me a revised version of the BASIC! They also sent a disk(!) asking that I send copies of any further programs causing trouble with the newest version. There was still a minor bug, and once more TSC solved it promptly and with dispatch I received a third version of the BASIC I had purchased. All of this work, it is to be remembered, went into a version that I am sure is NOT their best-seller -- I still use SWTPCo minifloppy running under miniflex -- although it is probable the fixes they were writing were applicable to all versions. Co-operation such as has been shown by the responsible firm of Technical Systems Consultants is all too rare in this day of minimal performance (or even pure scam). They should be recognized publicly for their attitude toward the customer and their pride in their product.

I own several versions of BASIC but only two have proved to be of practical application. They are the old, very slow, but reliable and accurate SWTPCo Disk BASIC Version 3.0 and the new TSC BASIC FOR 6800. The latter is the fastest of any BASIC I have ever seen run on any system. Even before using the built-in compiler and run-time package (yes, they are a part of the BASIC as furnished) the interpreter still outperforms any truthful timings I have seen of others, even those running on 780 systems at 4MHz while my system pokes along at 1 MHz. One must only become accustomed to slightly different modes of typing some expressions, doing a little more punctuating to gain a great deal more execution speed.

It must be a frightening experience to see the speed at which this BASIC will execute on a 6800 system running at 2 MHz!

Technical Systems Consultants have proved to be a reliable firm with excellent products and I unhesitatingly recommend them to the hobbyist looking for quality products at very affordable prices.

Thank you for your time.

John P. Tucker
John P. Tucker

P.S.

Don, if you decide to publish any of this and it is of a print quality suitable for your off-set process, please take your scissors to it and extract the meaning from the junk -- although that may take a major re-write instead!

John

December 2, 1979
1758 Comstock Lane
San Jose, Calif

Dear Don Williams

I wonder if anyone of your reading audience is interested in getting up a ARXX Amateur Radio Interest Group over the air?

I have had schedules over amateur radio with two ARXX radio amateurs in Australia for several years. We have been exchanging software for SSTV and RTTY. Our schedules are Friday/Saturday evenings (Calif.) on 21.260 KHz or 24.460 KHz at 01:00 or 02:00 GMT.

A few other ARXX users have been checking in from other countries (i.e. Panama and New Zealand). I have written a large amount of amateur radio software and have been exchanging it with the amateurs down under.

I'm sure some of your readers have amateur radio software they might like to exchange or discuss over the air?

Yours truly
Clay Abrams (K6AEP)

K. Panosian
12 Kismet Rd.
Bedford, MA 02025

Don Williams, Editor
'68' Micro Journal
3018 Hamill Rd.
PO Box 849
Hixson, TN 37343

Dear Don,

When a friend of mine, Norm Kahn, purchased a 400 (400) Extended Basic, I convinced him to use my clock board and clock routine to time the seven benchmark programs originally published in issue 10 of '68' Micro. The extended basic initially balked (Error #19) at the use of the UPR function, but after a call to APC on a Friday morning, Norm had a corrective procedure in hand and the following Monday. Computations are for the 400's fast response time. The table below summarizes all the times as have for the 400's series, including the original 400 timer for those benchmarks.

Benchmark #	1	2	3	4	5	6	7
6000 BASIC	14.9	24.7	96.1	105.3	109.8	174.1	204.5
6009 BASIC	1.1	3.5	10.99	11.3	11.95	17.59	26.7
6009 XTASIC	1.46	5.04	24.	26.56	27.26	34.85	45.61
6009 XTASIC	2.5	3.49	8.32	8.3	9.0	12.86	21.5

*The benchmarks were changed to use integer variables to obtain this last set of times. For example, benchmark #1 became:

400 FOR X=1 TO 1000
500 NEXT X

Note that the 6009 BASIC time for benchmark 1 is different than that I had previously reported. The current one is correct.

The problem with the UPR function and the XTASIC is that the base address of the function, in ROM, is version, is taken from End-of-User Memory - 4 was not as stated in the manual. For a 400 system, the UPR function address should be set in locations 77F9 and 77FA. The XTASIC being in locations 77F9 and 77FA. TSC informed Norm that this will be corrected in future releases of XTASIC.

R. Dembinski

Bob

**technical systems
consultants, inc.**

January 22, 1980

Don Williams
'68' Micro Journal
Williams Data-Comp Division
3018 Hamill Road
Hixson, TN 37343

Dear Don,

I enjoyed the chance to speak with you on the phone yesterday and appreciate your offer to inform your readers of our update policy.

For software packages listing under \$75.00, there is no guaranteed problem fixing. However, if a problem is properly reported we generally do supply patches to individuals who contact us.

For software packages listing over \$75.00 but without an offered maintenance schedule, fixes can be obtained. For a period of two months after purchase, the customer may obtain the latest version of the package free of charge. To do so he must return the original disk along with proof of purchase. Technical Systems Consultants will copy the latest version onto the disk and return it. If the customer has had possession of a package for over two months, he may still return his disk for a new copy (along with proof of purchase) but will be charged a \$10.00 handling fee. This policy applies only to corrections made to a particular product. New versions for a different DOS or application are not considered an update and must be purchased outright. If a user does not wish to return his original disk, Technical Systems Consultants will send a new disk with the latest version for a \$20.00 charge, so long as the user supplies proof of purchase.

Those software packages offered with a maintenance schedule are covered as specified in that maintenance agreement. These policies are subject to change at any time without notice.

I would also like to make a quick point about reporting problems. Telephone calls are an extremely poor method of reporting bugs in complex software packages. Ninety-five percent of such calls are a waste of time for both parties. The best solution is to thoroughly document the problem and MAIL it to us. This way we have a hard copy of exactly what is causing the problem and if it fails here, we have something concrete to work with. If one does call to report a bug, chances are he will be told to document and mail it to us.

Sincerely,

Don
Daniel E. Vanada

The Fifth Trenton Computer Festival TCF-80

April 19 & 20, 1980

10 AM to 6 PM
SATURDAY, 19th

at TRENTON STATE COLLEGE
Trenton, New Jersey

10 AM to 4 PM
SUNDAY, 20th



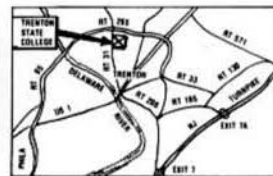
Super Outdoor Flea Market
Surplus computer gear, bargains galore, over 5 acres of space (RS/Spot, no electricity).



Indoor Commercial Exhibit Area
90 exhibitor booths showing newest products, special discounts, funky games to play.



Forums, Talks & Seminars
Meet the leading experts and hear sessions on robots, computer music, amateur radio, etc.



Convenient to NY, PA, MD & DEL.
Easy to get to, free parking for over 6,000 cars.

Hundreds of Door Prizes • Banquet Saturday Night

For additional information call 679-771-2457

Admission \$4 - Students \$2

Banquet \$10. Avoid disappointment - pre-register for the banquet! Send your check for \$10 per person to: TCF 80, Trenton State College, Trenton, N.J. 08605.

Sponsored by:
Amateur Computer Group of New Jersey
Philadelphia Area Computer Society
Trenton State College Digital Computer Society

Dept. of Engineering
Technology, Trenton State College
I.E.E.E., Princeton Section

COMPUTER SYSTEMS CONSULTANTS, INC.

1454 LATTI LANE, NW
CONERS, GEORGIA 30207
404-483-4570
EDGAR M. PASS, PH. D., PRESIDENT

December 26, 1979

Don Williams, Editor
'68' Micro Journal
3018 Howell Road
P.O. Box 849
Hixson, Tennessee 37341

Dear Don:

I have followed Jack Bryant's "Crunchers' Corner" with some interest. The installment of this article in the November/December 1979 issue has several problems. One is that the TRS/80 program has the following errors:

- on line 10 - array Q should have dimension 64, not 46;
- on line 38 - most BASIC's will give an error on an invalid MID range, and will not return a null string, as assumed here;
- on line 80 - H(L) should be H(N) in the first two statements and MID\$(S,1) should be MID\$(S,N,1)

The HP program appears to be correct, and does not have these problems. Apparently, the TRS/80 program was never tested. The other problem is that Step 1 of the explanation of the algorithm corresponds to line 80 of the TRS/80 program and contains the same error.

Sincerely,

E. N. Pass
E. N. Pass

Bill Vudall 1/80
2708 West Main
Olmsted, Minnesota 55406
Phone: (406) 337-3015

406-337-3015

December 28, 1979

68 Micro Journal
PO Box 849
3018 Howell Road
Hixson, Tennessee 37341

Dear Don:

In another attempt to continue my subscription to the Micro Journal I would like to submit the following fix for a fix. The improved MF-68 boot program presented helps but is not perfect. If a run-away program clobbers the MF-68 ISA it may not be configured properly to read the 1778 controller status at the start.

The following will correct this problem:

COMREQ EQU \$8018
CLR COMREQ

I hope this will help cure some hang-ups.

Sincerely yours,

Bill Vudall
Bill Vudall

December 31, 1979.
940 Evans Road
Knoxville, Tennessee 37904

Mr Don Williams Jr
'68' Micro Journal
3018 Howell Road
P. O. Box 849
Hixson, Tennessee 37341

Dear Sir:

One thing that would be of benefit to 8080 users would be a detailed, side-by-side discussion of DEC's FLEX and SBC's UCSD-B disk operating systems.

with emphasis on changing software written for one system to function on the other. Most people don't have access to both 8080s, and users of one system aren't familiar enough with the other to make the needed changes without help. I hope you can persuade one of your correspondents to provide an article to the '68' Micro Journal.

Very truly yours,

William K. Hensley
William K. Hensley

P.S. I'll send the printer driver listing as soon as I get the power supply for my 8080 rebuild - the primary winding of the power transformer shorted out!

WCH



Southwest Technical Products Corp

219 West Rhapsody
San Antonio, Texas 78216

For Immediate Press Release

SWTPC 6809 OPTIMIZING ASSEMBLER

SWTPC is proud to announce the availability of its 6809 Resident Optimizing Assembler. This assembler has been used extensively for in-house program development and has been a major improvement over previously available products. The assembler was designed from the ground up to facilitate modular block structured programming. It includes many features that assist in the rapid creation of error free programs with a minimum of effort. The significant features include arbitrary length symbols, procedures and data structures, local and public dictionaries, multiple program counters, automatic branch optimization, nested library file inclusion, exhaustive error checking and much more.

The assembler runs on any SWTPC 6809 disk based computer system with at least 24K of RAM memory. The program disk contains the resident 6809 assembler complete with a symbol table formatter overlay and utility programs to convert between binary disk files and paper tape. The ASM09 disk is available as either five or eight inch FLEX diskettes (please specify) for \$99.75 postpaid in the continental U.S.

1980 Newsletter -- #1

As we start a new year, we would like to bring everyone up to date on the latest products and new developments. 1979 saw the introduction of our 6809 computer systems and the rapid development of new software and peripherals to complement the system. In only about six months time a tremendous variety of software has been introduced for this system.

As we go into 1980, even more software is becoming available for our 6809 systems. UCSD Pascal is now available for the system. This form of Pascal is very popular in schools and colleges where portability of programs is important. The first package of application programs to be offered is also now available. See the enclosed sheet for descriptions. In addition to this, we are now ready to release our ASM09 Macro Assembler. This is probably the most advanced assembler on the market for the 6809 and anyone doing programs in assembler will certainly want a copy.

The first shipments of our new double-sided 5 1/4 inch disk drives will be made this month. The MF-69 uses two of the new double-headed Model 282 Siemens drives to give a total disk capacity of approximately 360K bytes. This makes it possible to use a small drive in many applications that were not practical previously due to the limited amount of storage.

Southwest Technical Products now has a contract to distribute Centronics printers with our systems. We will continue to make the Qume Sprint 3 daisy wheel printers available with our systems, but will no longer offer the 6540 and the 8300 printers. Centronics has a very large variety of printers and can provide local service in many parts of the country. They will also allow us to pass on to you a 90 day warranty on their machines. Normal discounts will apply on all Centronics printers.

I am sure that most of you are aware by now that we are putting all 6809 computers in the same type cabinets. The smaller 8K and 56K systems have a new motherboard that allows use of our new dual I/O cards. The "boilerplate" one-inch alloy aluminum covers and bottoms, as well as the cooling fan, are now standard items on these products. The introduction of a 6809 kit again provides an inexpensive way for people to build up a system with minimum initial investment.

The version of FLEX-9 now being shipped with all disk systems includes a 6809 version of our EPROM software for use with the MP-A EPROM programmer. The 8" version utility that allows you to use the new cleaning disc with your system. FLEX-9 also includes four types of printer drivers for the various types of printers that may also be attached to the system.

The last new item is our new S-32 universal memory card. This memory card makes it possible to put together a system with exactly the amount, and EPROM/ROM/RAM mixture of memory that you need. 2716 EPROM's and pin compatible RAM's and ROM's may be intermixed throughout the board in as small as 4K blocks. This is perfect for process control applications, or situations where it is desirable to have programs stored in ROM. You are free to buy the amount and mixture that you need, up to the 32K byte capacity of the board.

Southwest Technical Products Corporation
219 West Rhapsody
San Antonio, Texas 78216
(512) 344-0241

Applevalley Day School, Inc.

Offering Our Own Business Software
in SWTPC Disk Ver. 3.0

PROGRAMMER:
RICHARD G. CAGLE

11103 Sagepark Lane
Houston, Tx. 77089
713/481-3586



technical systems
consultants, inc.

***** FLASH *****
January 9, 1980

6809 FLEX Text Processing System

As many of you know, we are developing a new 6809 text processor which supports proportional spacing (on a character level) as well as all the other features of Qume and Diablo type printers such as subscripts, superscripts, reversed line-feeds, etc. Unfortunately, it appears that it will be some time before that package is completed. Due to this fact plus the extreme demand for such, we are releasing a re-done version of our greatest 6800 Text Processing System. Unlike most of our other 6800 products, it is not position-independent or reentrant.

As is our normal policy, there will be no upgrade price if a customer purchases the current 6809 text processor now and later wishes to purchase the proportional spacing processor. One point that might be made is that there will be no great advantage to the planned text processor if the customer does not have a printer with proportional spacing ability (such as the Qume or Diablo).

The 6809 Text Processing System is available on 5 1/4 or 8 inch FLEX disks only. There is no cassette version and the source listing is not included. The package is part number SP09-13 and retails for \$60.00.

Quantity	Price
1	\$60.00

If you have any questions regarding this item, give us a call at (317)463-2502 or Telex 276143.

Technical Systems Consultants, Inc.



technical systems
consultants, inc.

The UnifLEX Operating System

I. General Information

Technical Systems Consultants Inc., is in the final development stages of the new operating system, UnifLEX (UnifLEX is a trademark of Technical Systems Consultants Inc.). The system's design has been influenced primarily by two other operating systems, FLEX* and UNIX* (UNIX is a trademark of Bell Laboratories). UnifLEX retains the flexibility and ease of use of FLEX while incorporating some of the widely accepted structures of UNIX.

UnifLEX is a true multi-tasking, multi-user operating system. It supports a hierarchical file system allowing file sizes up to one billion bytes and disk capacities of over eight billion bytes. All files are fully protected. A user may read, write, or execute protect files on an individual basis. All system I/O is device independent since I/O devices and files are treated in an identical fashion. Any combination of interrupt driven devices may be attached to the system. A sample system configuration might have a ninety megabyte disk, a thirty megabyte disk, two eight inch floppies, a parallel driven printer, and six user terminals. Any running task may initiate another task in an asynchronous manner. Inter-task communication is also supported. Task swapping may take place on those systems incorporating an appropriate swapping device. In real time applications, where swapping may be detrimental, it is possible to lock a task in main memory.

The user's main connection with the system is through the command language. This is the vehicle for inputting commands to the system. It supports a wide variety of features including parameter passing, command variables, control flow primitives, and various types of name matching. Input to the command language may be from the terminal or from any file (including other devices). This enables very complex command files to be created and run at will.

The overall design of UnifLEX is geared toward the larger microcomputer systems. It is our opinion that small systems deserve small, compact operating systems (such as FLEX) while larger systems require much more sophistication. We will continue to support FLEX one hundred percent since the small micro is and will be the predominant system for quite some time. UnifLEX will be for the upcoming generation of microcomputers which support massive amounts of main and secondary memory. It will be the operating system we will not only provide for the 6809 but also all 68000 based machines.

II. Hardware Requirements

UnifLEX will require a system based on either the 6809 or 68000. No attempt at a 6800 version will be made. The mainframe must support extended addressing (a minimum of 19 address lines) and some form of memory segment management. The memory mapper must have the minimum ability of mapping memory segments no larger than 4k, anywhere in the cpu address space. The operating system will support multiple mapping registers as well as hardware memory protection schemes although neither is a requirement. A minimum of 96K of memory is required.

All system I/O devices must be interrupt driven. Terminals may be run either from separate ACIA type controllers or through an I/O buffer processor. At least one eight inch, single density floppy disk drive must be supported. The drive must be controlled with a DMA, interrupt producing controller. This drive will be used for program transportation and system booting. Any other disk controllers may be added to the system but they should be of the DMA variety.

III. System Support

The basic UnifLEX system will include the operating system, several system utilities, a text editor, macro assembler, and system configuration programs. System maintenance will be available as an option. Each copy of UnifLEX sold will include a single cpu license and will have a unique serial number. Any additional software purchased for the system will be required to have a matching serial number for proper operation. System maintenance, if purchased, will provide updates on eight inch floppy diskette. Updates will consist of fixed 'bugs' as well as added system enhancements. Those not obtaining maintenance will only be informed of bugs for a period of 30 days.

There is currently a wide variety of support software in the development stages to run under UnifLEX. These include a C compiler, Pascal, Text Processing system, a debug package, and BASIC incremental compiler which is compatible with our Extended BASIC. Many other programs are being considered, including FORTRAN, RPG/II, and a data base management system.

IV. Final Comments

We will not claim that UnifLEX is the 'ultimate' operating system. It is impossible to create a program such as an operating system with which no one can find fault. We do feel, however, that UnifLEX does represent the state-of-the-art in operating systems and incorporates as many 'large system' features as is practical without losing the friendliness of the smaller system.

COMPUTERWARE
1512 Encinitas Blvd
ENCINITAS, CALIFORNIA 92024

January 15, 1980

Don Williams
'68 Micro Journal
3018 Hamill Road
Bixen, Tenn. 37343

Dear Don,

As you know Computerware has been developing a very extensive library of 6800/6809 application software for the home, small business, and commercial user. We currently have implemented Accounts Receivable, Accounts Payable, Payroll, Inventory Control, General Ledger, Cash Flow, Mailing System, Real Estate Package, and a Random File Organizer/Reporting System. Of course, the future will introduce many more packages currently in work.

To date we have limited our applications to use with the Smoke Signal Broadcasting DOS. However, we have been approached by many FLEX users who would like to use our software as well. We are therefore considering undertaking the project of creating FLEX compatible packages.

We would very much appreciate the response of your readers who may have an interest in seeing our software offered in a FLEX compatible format. Our decision on this project will depend largely on the demand from the 6800/6809 market place (both user and dealer) in the form of letters, inquiries, and commitments.

May I again thank you and your team at '68 Micro Journal for providing this media of forum and exchange to the 68XX community.

Sincerely,
Paul Searby
Paul Searby
Computerware

*FLEX is the trademark of TSC.



Dynasoft Systems has introduced a complete PASCAL program development system for the 6800 microprocessor. DYNASOFT PASCAL is a subset of standard PASCAL intended for small and medium-sized cassette-based systems which cannot support full-scale implementations such as UCSD PASCAL. It includes most of the control structures of standard PASCAL and supports the data types CHAR, INTEGER, boolean, scalar, subrange, pointer and ARRAY. Language extensions include EXTERN (machine language) PROCEDURES and FUNCTIONS, LINK to other PASCAL programs and an optional OTHERWISE clause on the CASE statement.

The one-pass compiler produces fast, relocatable, ROMable pseudo-code which requires only a 1300 byte interpreter to execute, making it possible to run small programs in as little as 2K bytes. The entire system, which includes the compiler, interpreter, a line-oriented editor, and system supervisor, occupies less than 8K bytes and will compile a 2000 character source program in 12K bytes of memory.

The system is designed to operate on cassette-based systems such as the SWTPC 6800 which use MINBUG, SWTBUG, or equivalent monitors, but all I/O calls are made through common points in the interpreter, making it simple to adapt for most other configurations. The standard version requires contiguous RAM starting at \$0000 but it can be supplied on special order in ROMable form starting at \$C000. A cross-compiler is also available which runs on the CDC CYBER series of computers under PASCAL 6000, version 3.

Price for the basic cassette version, with manual, is \$35.00. For further information, contact Dynasoft Systems, P.O. Box 51, Windsor Junction, N.S., Canada B0N 2V0.

NEW PRODUCT

For the 6809 user who has applications programs running under the SWTPC BASIC version 3.5 and needs time to changeover to newer BASIC formats, Omnitronics Inc., 1897 Rt. 33, Concord Sq, Hamilton Sq, NJ 08690, announces a completely source compatible BASIC-0935 running on all 6809 S50 bus computers, running FLEX™. This version will accept most all SWTPC BASIC disk versions without modification.

No manual is included. It is assumed that all users have the original SWTPC manual but included with BASIC-0935 is instructions for the extensions to the standard SWTPC BASIC. The extensions include: Complete ERROR trapping, AUTOMATIC line number insertions, renumbering (very fast) of line numbers, SET - RESET, these commands turn on and off graphics pixels (X,Y addressable terminal), double variables, string variables such as a\$-z\$ may be legally DIM'ed, run binary programs from BASIC-0935 (example: +TTYSET PS=N - set pause off in FLEX™), I/O has been configured to a more standard - no parity, 1 stop bit, 8 data bits, ACIA assumed on control port at \$E004 (can be changed to your control port address), ERROR trapping includes new error number (37) indicating binary program invoked by + operator, was not found on the disk.

BASIC-0935 runs approximately 40% faster and honors FLEX™ 'End of Memory' pointers. It is monitor independent requiring only 09 FLEX™.

BASIC-0935 is optimized 6809 code, running in slightly over 12K of memory and is not relocatable. EPROM (2716) version are available on special order, at any memory location required.

Additional enhancements are being developed and will be announced when completely debugged, updates will be available at a modest cost.

ACCOUNTS RECEIVABLE

for the FLEX OPERATING SYSTEM

GREAT PLAINS COMPUTER CO has converted the Osborne Accounts Receivable program to operate under the Technical Systems Consultants 'FLEX' Operating System. Thus TAMJ and SOUTHWEST TECHNICAL PRODUCTS owners who have FLEX now have available a powerful, complete minicomputer quality ACCOUNTS RECEIVABLE package. The programs were rewritten in TSC's 'XBASIC' and operate using keyed random access files (ISAM). The system is fast and forgiving.

The ACCOUNTS RECEIVABLE package contains three features not included in the original Osborne or in most conversions.

1. A keyed random access file system (ISAM) maintains the speed and power of the original WANG minicomputer version.
2. The user is always 'backed up' against failure because the system issues step-by-step prompts. Backu can be skipped if the user explicitly instructs the system to bypass it.
3. Password protection was extended to help prevent unauthorized access to the Accounts Receivable programs and data files.

SCHEDULE

ACCOUNTS RECEIVABLE	now shipping
GENERAL LEDGER	JAN 31, 1980 (now accepting orders)
ACCOUNTS PAYABLE	available FEB 28, 1980
PAYROLL	estimated MAR 31, 1980
COST ACCOUNTING for PAYROLL	estimated April 30, 1980

PRICES

Each of the above 5 programs is priced the same:

Compiled program	\$ 100. (single machine end user)
Source program	\$ 300. (single user)
License fee	\$400. (non exclusive dealer)

TERMS

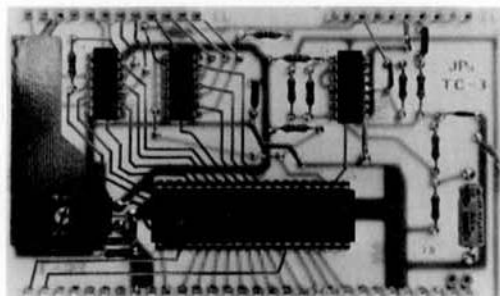
Source code will be sold to qualified Dealers and OEM companies. Software agreements will be sent for your signature upon request.

Payment conditions are:

CASH --- C.O.D. --- Previously approved credit application

JPC PRODUCTS FOR

6800 COMPUTERS



High Performance Cassette Interface

- **FAST** - 4800 Baud Loads 4K in 8 Seconds!
- **RELIABLE** - Error Rate Less Than 1 in 10⁹ Bytes.
- **CONVENIENT** - Plugs Directly Into The SWTPC.
- **PLUS** - A Fully Buffered 8 Bit Output Port Provided.
- **LOW COST** - \$59.95 For Complete Kit.
- **OPTIONAL** - CFM/3 File Manager.
Manual & Listing \$19.95
For Cassette Addl \$ 6.95

TERMS: CASH, MC or VISA; Shipping & Handling \$3.00



Order Phone (505) 294-4623
P.O. Box 5615
Albuquerque, N.M. 87185



Inventory Problems?

Are you having trouble keeping the right nuts and bolts in stock? Since even a simple mistake can cost you time and money, a good inventory system should do more than just count parts. It should tell you exactly what you need, when you need it, where to get it, and how much it will cost.

The MSI Inventory System Seven enables you to maintain a versatile data base for controlling inventory. It lists part number, description, quantity on hand, vendor, cost, selling price, optional pricing, usage levels for previous month, present month, and year-to-date, and much more.

When quantity on hand items reach minimum levels, the System Seven compiles an automatic reorder list. This list can be generated by specific vendor as well as a complete listing of all materials to be ordered.

In addition to the item listing, the Inventory System Seven "bill of materials" provides you with a complete inventory of items used in the manufacture of subassemblies and complete products. It also contains other cost items such as labor costs, total raw materials costs, and miscellaneous costs.

The MSI Inventory System Seven is built around the versatile MSI 6800A Computer with 56K of RAM. An integral dual mini-floppy memory gives you an additional 630K of memory and makes

inventory control fast and efficient. The System Seven will interface with any industry standard CRT, and you have the option of both a "daisy wheel" word processor for high quality document preparation and a dot matrix printer for high speed production.

The System Seven can be expanded to handle all your data processing needs or you can select one of nine other MSI systems now available for business, industrial, scientific, educational, and personal applications.

If you need more than just a nuts and bolts inventory system, we have more information about how the Inventory System Seven can solve your problems economically.



MSI Inventory System Seven

MSI

Midwest Scientific

220 W. Cedar, Olathe, Kansas 66061, (913) 764-3273
TWX 910 749 6403 (MSI OLAT), TELEX 42525 (MSI A OLAT)



A Complete Library of M6800 System Software!



These books will help you code, assemble, link, load, relocate, and debug your 6800 assembly language programs. Each book includes complete source code, object code in hexadecimal format, and machine readable (PAPERBYTE®) bar code listings, to make entering and modifying the programs as easy as possible.

Tracer: A 6800 Debugging Program is for the programmer looking for good debugging software. Tracer features single step execution using dynamic break points, register examination and modification, and memory examination and modification. This book includes detailed Tracer program notes and a reprint of "Jack and the Machine Debug" (from the December 1977 issue of BYTE magazine).

ISBN 0-931718-02-3 Pages: 24 Price: \$6

Authors: Robert D. Grappel & Jack E. Hemenway

MONDEB: An Advanced M6800 Monitor-Debugger has all the general features of Motorola's MIKBUG monitor as well as numerous other capabilities. Some of the command capabilities of MONDEB include displaying and setting the contents of registers, setting interrupts for debugging, testing a programmable memory range for bad memory locations, changing the display and input base of numbers, displaying the contents of memory, searching for a specified string, copying a range of bytes from one location in memory to another, and defining the location to which control will transfer upon receipt of an interrupt.

ISBN 0-931718-06-6 Author: Don Peters Pages: 88 Price: \$5

RA6800ML: An M6800 Relocatable Macro Assembler is a two pass assembler for the Motorola 6800 microprocessor. The Assembler can produce a program listing, a sorted Symbol Table listing and relocatable object code. The object code is loaded and linked with other assembled modules using the Linking Loader LINK68. There is a complete description of the 6800 Assembly language and its components. Each major routine of the Assembler is described in detail, complete with flow charts and a cross reference showing all calling and called-by routines, pointers, flags, and temporary variables. In addition, details on interfacing and using the Assembler and error messages generated by the Assembler are included. This book provides the necessary background for coding programs in the 6800 assembly language, and for understanding innermost operations of the Assembler.

ISBN 0-931718-10-4

Author: Jack E. Hemenway

Pages: 164 Price: \$25

LINK68: An M6800 Linking Loader is a one pass linking loader which allows separately translated relocatable object modules to be loaded and linked together to form a single executable load module, and to relocate modules in memory. It produces a load map and a load module in Motorola MIKBUG loader format. This book provides everything necessary for the user to easily learn about the system, including a detailed description of the major routines of the Linking Loader, including flow charts. While implementing the system, the user has an opportunity to learn about the nature of linking loader design as well as simply acquiring a useful software tool.

ISBN 0-931718-09-0

Authors: Robert D. Grappel & Jack E. Hemenway

Pages: 72 Price: \$8

Tiny Assembler 6800, Version 3.1 is a small (4 K) but sophisticated and useful assembler for a large subset of the Motorola 6800 assembly language. The book includes detailed notes on the design and implementation of Version 3.0 of the assembler, a complete description of the enhancements upgrading the Tiny Assembler to Version 3.1, an updated user's guide, and complete listings for both versions, making this book the most complete documentation possible for Jack Emmerich's Tiny Assembler.

ISBN 0-931718-08-2 Pages: 80 Price: \$9

Author: Jack Emmerichs

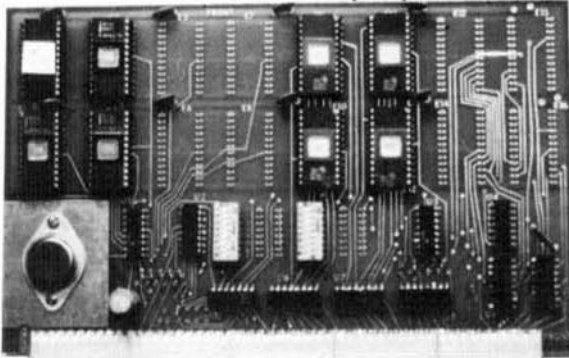
Please send		<input type="checkbox"/> _____ copies of RA6800ML: An M6800 Relocatable Macro Assembler <input type="checkbox"/> _____ copies of LINK68: An M6800 Linking Loader <input type="checkbox"/> _____ copies of Tracer: A 6800 Debugging Program <input type="checkbox"/> _____ copies of MONDEB: An Advanced M6800 Monitor-Debugger <input type="checkbox"/> _____ copies of Tiny Assembler 6800, Version 3.1	
Name _____		Title _____ Company _____	
Street _____	City _____	State/Province _____	Code _____
<input type="checkbox"/> Check enclosed in the amount of \$ _____ <input type="checkbox"/> Bill Visa <input type="checkbox"/> Bill Master Charge Card No. _____ Exp. Date _____ Add 60¢ per book to cover postage and handling.			
BYTE BOOKS B30		70 Main Street Peterborough NH 03458 	

Circle 35 on inquiry card.

DIGITAL SERVICE & DESIGN

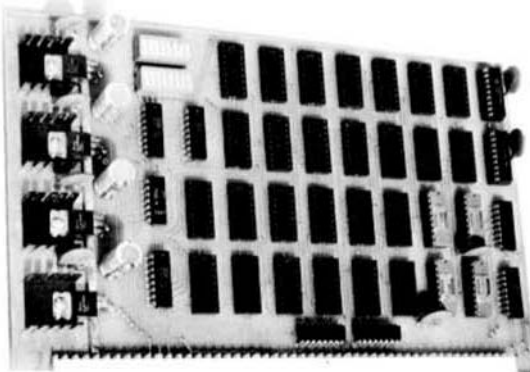


P.O. BOX 741
NEWARK, OHIO 43055
Phone (614) 366 6314



DSD P/R-32K\$27.00

32K or 16K EPROM & RAM memory card.
2716 2K x 8 or 2758 1K x 8 5V only EPROMS.
TMS 4016 2K x 8 or MK 4118 1K x 8 5V RAMS.
Up to 4 independent addressed 8K blocks. Dip
switch or jumper selected. Size 9" x 5 1/2"



DSD 2114-16K\$27.00

Full Static 16K Ram memory card designed to
use the 2114 or TMS 4045 1024 x 4 Static Ram.
The card has two independent addressed 8K
memory blocks. Card size 9" x 5 1/2". Power
requirements 7-8V unreg. @ 3.5A.

DSD U P 8255M\$14.00

Universal parallel interface card with wire wrap
area using INTEL'S 8255 parallel peripheral
interface chip. 24 programmable I/O lines.
(Three 8 bit Ports or Two 8 bit Ports with hand-
shaking) Card size 5 1/2" x 5" Standard SS-50 30
pin I/O BUS. 5V only.

Cards are bare with data and edge connector.
Ohio residents add 4 1/2% sales tax.

6847 Color Graphic card in design

MULTI-USER ON A 56K 6809

Support four totally independent terminals with one computer.
Each terminal has access to the services you need most in
your classroom or business.

COMMON PILOT - the best CAI system in the world. useful in any
application where computer-user dialogue is
desirable.

STUDENT BASIC - real numbers, functions, arrays, strings, loops,
subroutines, and random disk-file access.

SCREEN-ORIENTED EDITOR - edit lines directly by over-
typing, inserting, and deleting; useful for
programming and word processing.

Hardware requirements: SwTPC /09 (the cheaper 6809), 56 K,
MP-N calculator board, dual floppy disks (either size), one to
four MP-S serial interfaces, one to four SwTPC CT-62 terminals.

\$250.00 single copy (specify 5" or 8")
(quantity discounts available)



6800 PASCAL

DYNASOFT PASCAL is a cassette based
PASCAL subset designed to run on most 6800
systems with 12K or more of memory.

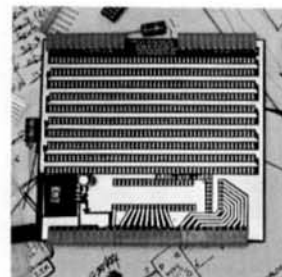
DYNASOFT PASCAL includes most of the control
structures of standard PASCAL including IF-THEN-ELSE,
CASE-OF-OTHERWISE, WHILE-DO, REPEAT-UNTIL, FOR-
TO/DOWNTODO, and recursive PROCEDURE's and
FUNCTION's. It supports the data types INTEGER,
CHAR, BOOLEAN, scalar (user-defined), subrange,
pointer and ARRAY. It is built around a one pass
compiler which produces fast, compact p-code and
comes complete with a line-oriented text editor, p-
code interpreter, and program SAVE and LOAD
routines. The whole system resides in less than 8K
and is also available in ROM.

The cassette version is priced at \$35 plus \$3 for
postage and handling. Satisfaction guaranteed.



P.O. BOX 51
WINDSOR JCT., N.S.
CANADA B0N 2V0
(902) 861-2202

THE MICRO P.O. BOX 1110, DEL MAR, CA 92014 714-942-2400



Software? A stripped down version of our U2708 Utility program was published in *Dr. Dobb's Journal of Computer Calisthenics & Orthodontia*, February issue, #32. Or the complete version of U2708 is available in the B-08 Owner's Manual for only \$10.00.

BUT...

If you're not a dyed in the wool wherwrapper and would like to purchase

The Real Thing

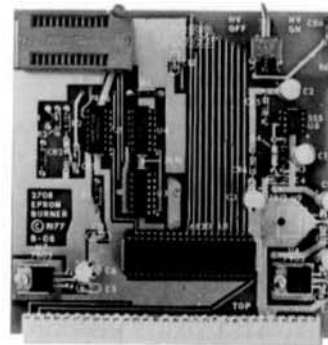
Assembled, Tested, Burned in and

★ READY TO GO ★

GOTCHA COVERED

B-08 Eprom Programmer \$99.95

U2708 Software in Eprom \$29.95



CURSOR/SCREEN EDITING IS HERE FOR THE SS-50!

Research conducted by a reliable source (MY WIFE) indicates that you spend more time looking at your CURSOR than your own FAMILY! Don't you think it's about time you used the video board and the cursor which cost you good money for something other than a poor excuse for not looking at your family?!

The unique SCREDITOR CURSOR-BASED SCREEN EDITING SYSTEM is just what you need to make your hardware work! Take a look at a few of the features of the SCREDITOR and see what you're missing —

- * EVERYTHING HAPPENS ON THE SCREEN — YOU SEE WHAT IS HAPPENING AS IT HAPPENS!
- * INSERT CHARACTERS, WORDS, LINES OR EVEN OTHER FILES, AND WATCH IT AS IT HAPPENS — RIGHT ON THE SCREEN!
- * FILE SIZE IS ONLY LIMITED BY DISC SPACE!
- * PRINT COMMAND PROVIDES HARDCOPY WITH ADJUSTABLE MARGIN TO MATCH YOUR PRINTER!
- * AUDIBLE FEEDBACK THROUGH A USER-SUPPLIED NEWTECH BOARD SOUNDS KEYSTROKES AND EDIT-MESSAGES AS THEY OCCUR!
- * SELECTABLE TEXT MODE ALLOWS CONTINUOUS TYPING WITHOUT CONCERN FOR MARGINS!
- * FULLY UTILIZES THE FLEXIBILITY AVAILABLE ONLY IN A MEMORY-MAPPED VIDEO DISPLAY!
- * MOVE LINES AND PARAGRAPHS FROM PLACE TO PLACE IN THE FILE YOU ARE EDITING — EVEN BEYOND THE BLOCK IN MEMORY!
- * SET, DISPLAY, CLEAR AND USE TABS LIKE A TYPEWRITER!
- * FIND AND CHANGE COMMANDS WAIT TO BE VERIFIED TO HELP STAMP OUT ERRORS!
- * IMPORTANT INFORMATION SUCH AS LINE, COLUMN, MODE, ETC. . . , IS CONTINUOUSLY DISPLAYED AND UPDATED ON THE SCREEN!
- * EDITING OPERATIONS ARE CURSOR-ORIENTED. THE CURSOR POINTS TO WHERE THE ACTION IS!

The features go on and on! With FOURTEEN major commands, TWO edit modes, TWO major file handling modes and TWENTY-TWO screen operators, the SCREDITOR offers a level of control and convenience never before available to users of the SS-50 buss.

Order your copy of the SCREDITOR for 16 x 64 displays (such as the Thomas Instrumentation Board) OR for 24 x 80 displays (such as SSB VOB-1) to run under SSB DOS68.51X today! To place your order, call or write —

BE SURE to specify the version desired when ordering.

ALFORD AND ASSOCIATES
P. O. BOX 6743
RICHMOND, VA. 23230
804-329-3906

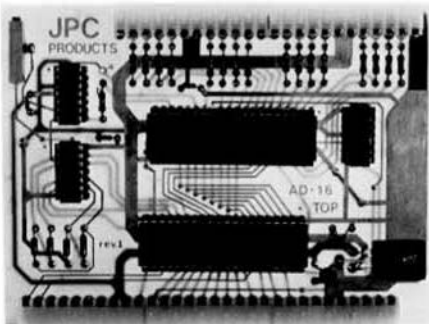
\$99.95

Va. residents add 4%.

VISA and M. C. accepted graciously. Sorry, no COD's. Shipment is normally within three days of receipt of order unless by check. 6809 version available soon for SSB's 6809 DOS.

JPC PRODUCTS FOR

6800 COMPUTERS



USES
ONE
I/O
SLOT

16 CHANNEL A/D BOARD

- 8 BIT DATA
- SOFTWARE CONTROLLED GAIN
- 3300 SAMPLES PER SECOND
- $\pm 0.7\%$ ACCURACY

COMPLETE KIT: AD-16 \$69.95

Terms: Cash, MC or Visa; Shipping & Handling \$3.00



Order Phone (505) 294-4623
P.O. Box 5815
Albuquerque, N.M. 87185

Model EP-2A-88 EPROM Programmer



Fast as Jackrabbits . . . Well, almost!

In Australia, two rabbits can reproduce over 13 million offspring in 3 years . . . at 105 seconds per 2716, the EP-2A-88 can reproduce 1,892,160 EPROMS in 3 years. Single push button control, the EP-2A-88 checks if EPROMS are erased, programs and verifies. It also checks for defective EPROMS.

Two basic models are available. The EP-2A-88-1 will accept Copy (CM) modules for the 2758, and 2716 EPROMS. The EP-2A-88-2 will accept copy modules for the 2716, 2732 and TMS 2532 EPROMS. Power requirements are 115 VAC 50/60 Hertz at 15 watts.

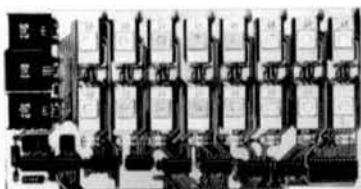
Part No.	Description	Price
EP-2A-88-1	EPROM Programmer	\$450.00
EP-2A-88-2	EPROM Programmer	\$500.00
CM-51	Copy Module for 2716, TMS 2516 EPROMS	25.00
CM-71	Copy Module for 2758 EPROMS	25.00
CM-21	Copy Module for 2732 EPROMS	25.00
CM-41	Copy Module for TMS 2532 EPROMS	25.00
	Non Standard Voltage Option (220v, 240v, 100v)	15.00

Optimal Technology, Inc.
Blue Wood 127, Earlyville, Virginia 22936
Phone (804) 973-5482

DIGITAL RESEARCH COMPUTERS

(214) 494-1505

16K EPROM CARD-S 100 BUSS



\$59.95
KIT

FIRST TIME OFFERED!
BLANK PC BOARD - \$28

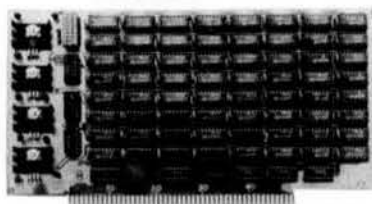
USES 2708's!

Thousands of personal and business systems around the world use this board with complete satisfaction. Puts 16K of software on line at **ALL TIMES!** Kit features a top quality soldermasked and silk-screened PC board and first run parts and sockets. Any number of EPROM locations may be disabled to avoid any memory conflicts. Fully buffered and has WAIT STATE capabilities.

OUR 450 NS 2708'S
ARE \$8.95 EA. WITH
PURCHASE OF KIT

ASSEMBLED
AND FULLY TESTED
ADD \$25

8K LOW POWER RAM KIT-S 100 BUSS SALE



PRICE
CUT!

\$119.50
KIT

21L02
(450 NS RAMS!)

Thousands of computer systems rely on this rugged, work horse, RAM board. Designed for error-free, NO HASSLE, systems use.

Blank PC Board w/Documentation
\$29.95

Low Profile Socket Set...13.50
Support IC's (TTL & Regulators)
\$9.75

Bypass CAP's (Disc & Tantalums)
\$4.50

ASSEMBLED AND FULLY
BURNED IN ADD \$30

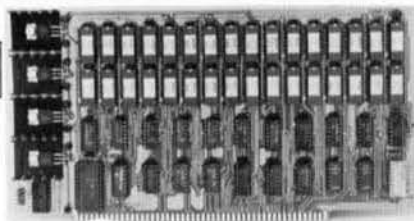
ALL ASSEMBLED BOARDS
ARE TESTED AT 4MHZ.

16K STATIC RAM KIT-S 100 BUSS

PRICE CUT!

\$259 KIT

FOR 4MHZ
ADD \$25



KIT FEATURES

1. Addressable as four separate 4K blocks
2. ON BOARD BANK SELECT circuitry (Cromco Slendergit). Allows up to 512K on line
3. Uses 2114 (450NS) 4K Static Rams
4. ON BOARD SELECTABLE WAIT STATES
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers
6. All address and data lines fully buffered
7. Kit includes ALL parts and sockets
8. PHANTOM is jumpired to PIN 67
9. LOW POWER, under 1.5 amps TYPICAL from the +5 Volt Bus
10. Blank PC Board can be populated as any multiple of 4K

BLANK PC BOARD W/DATA-\$33
LOW PROFILE SOCKET SET-\$12
SUPPORT IC'S & CAPS-\$19.95
ASSEMBLED & TESTED-ADD \$30

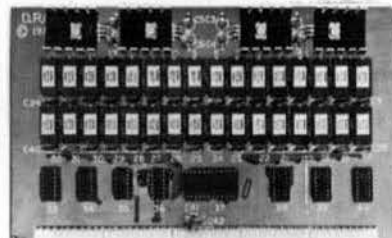
OUR #1 SELLING
RAM BOARD!

16K STATIC RAM SS-50 BUSS

PRICE CUT!

\$249 KIT

FULLY STATIC
AT DYNAMIC
PRICES



FOR SWTPC
6800 BUSS!

ASSEMBLED AND
TESTED - \$30

KIT FEATURES

1. Addressable on 16K boundaries
2. Uses 2114 Static Ram
3. Runs at Full Speed
4. Double sided PC Board Solder mask and silk screened layout Gold fingers
5. All Parts and Sockets included
6. Low Power Under 1.5 Amps Typical

BLANK PC BOARD-\$26 COMPLETE SOCKET SET-\$12
SUPPORT IC'S AND CAPS-\$19.95

PROC. TECH. QUITS THE MICROPROCESSOR BUSINESS!
FACTORY CLOSE OUT - SPECIAL PURCHASE!
#16KRA

16K S-100 Dynamic Ram Board

\$149.95

We purchased the remaining inventory of FET's for 16K Ram Board when they recently closed their plant. Don't miss the boat! These are brand new, fully tested, ASSEMBLED and ready to go. All are sold with our standard 90 day limited warranty!!
Orig. \$429 each! 72 Page Full Manual, Included Free!

Z-80 PROGRAMMING MANUAL

By MOSTEK, or ZILOG. The most detailed explanation ever on the working of the Z-80 CPU CHIPS. At least one full page on each of the 158 Z-80 instructions. A must reference manual for any user of the Z-80, 300 pages.

\$12.95

SALE! LOW POWER - 250NS 8 FOR
2114 RAM SALE! \$55

4K STATIC RAM'S. MAJOR BRAND. NEW PARTS.
These are the most sought after 2114's. LOW POWER and 250NS FAST.
SPECIAL SALE: \$7.99 ea. or 8 For \$55

Digital Research Computers
(OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 494-1505

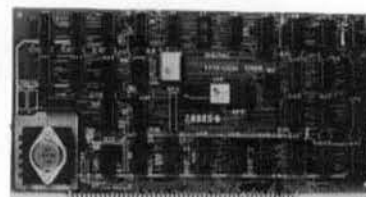
Perfect for
OEM's

S-100 Z80 CPU CARD

\$159.95

WIRED!
NOT A KIT!

4 MHZ



ASSEMBLED AND TESTED! READY TO USE! Over 3 years of design efforts were required to produce a TRUE S-100 Z80 CPU at a genuinely bargain price!

BRAND NEW!

FEATURES

- 2 or 4 MHz Operation
- Generates MWRITE, so no front panel required.
- Jump on reset capability
- 8080 Signals emulated for S-100 compatibility.
- ★ Top Quality PCB. Silk Screened, Solder Masked, Gold Plated Contact Fingers.

NEW! G.I. COMPUTER SOUND CHIP

AY3-8910. As featured in July, 1979 BYTE! A fantastically powerful Sound & Music Generator. Perfect for use with any 8 Bit Microprocessor. Contains: 3 Tone Channels, Noise Generator, 3 Channels of Amplitude Control, 16 Bit Envelope Period Control, 2-8 Bus Parallel I/O, 3 D to A Converters, plus much more! All in one 40 Pin DIP. Super easy to interface to the S-100 or other busses.

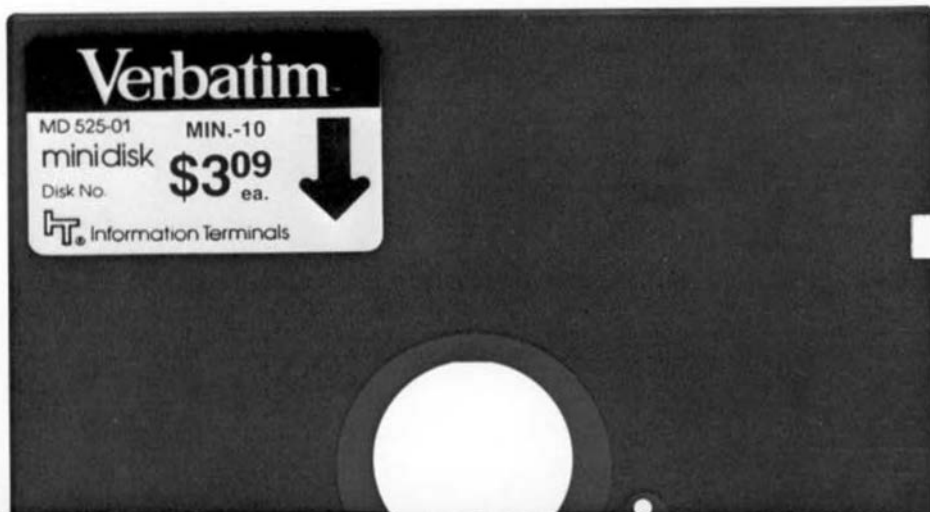
SPECIAL OFFER: \$14.95 each Add \$3 for 64 page Data Manual.

TERMS: Add \$1.00 postage, we pay balance. Orders under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. 90 Day Money Back Guarantee on all items.

NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

5-1/4" Minidisk — Soft or Hard Sector

Dealer and Volume Discounts Available



S
A
V
E

D
I
S
K

DISK

5" Soft Sector \$3.09 each
5" 10-16 Sector \$3.09 each
8" Single Side, Double
Density \$4.50 each
8" Double Side, Double
Density \$6.50 each
5" Plastic Box-plus \$3.00
Minimum Order 10 (1 box)

minidisk™
HT

Verbatim™

SOUTH EAST MEDIA SUPPLY

P.O. Box 794

615-870-1993

Hixson, TN 37343

H H H Enterprises
Box 493, Laurel, Md.
20810 301-953-1155

ANNOUNCING!!! GIMIX MEMORY!!!

32K OF STATIC RAM

32K OF STATIC RAM on one card WITH, extended addressing (to 1Meg), fully socketed, any 8K segment can be defeated, fully dip switched, all gold contacts and the QUALITY of GIMIX.

32 K FOR ONLY \$548.15

24 K FOR ONLY \$438.14

16 K FOR ONLY \$328.12

All sizes come with ALL sockets so just buy what you need and fill in the blanks with 2114's latter.

SOFTWARE CONTROL BOARDS

16 K ram with software control of 4 ea. 4K blocks for address, write protect and phantom.

16K GHOSTABLE ONLY \$368.16

See GIMIX Ad on page 3



KING OF THE HILL

VIDEO BOARD is the ONLY one that has software control of character generators and half/full intensity, with reverse and graphics and RAM character set. You can load character sets from your disk or tape under program control!!!!

VDM76 (GHOSTABLE) ONLY \$458.76

GIMIX MAINFRAME

32K with choice of I/O card, a super CPU, fan, KEY switch, BIG power, 15 slot mother board, DIP switch readdressing.

SYSTEM 49 ONLY \$1549.49

Also — with the system, order the VDM+, a Cherry Kbd., 12 inch monitor, and all cables for only \$810 extra, and beat any terminal.

SYSTEM 49+ + ONLY \$2359.49

H H H ENTERPRISES

BOX 493, Laurel, MD.
ZIP 20810
PHONE 301-953-1155

ED SMITH'S SOFTWARE WORKS 6809 SOFTWARE TOOLS

RRMAC M6809 RELOCATABLE RECURSIVE MACROASSEMBLER. The one assembler that contains real macro capabilities (see our NOV/DEC AD). RRMAC is designed with the assembly language programmer in mind and contains many programmer convenience features. RRMAC contains a mini-editor, supports spooling or co-resident assembly, allows insert files, is romable, generates cross-references, execution times, lists target addresses of all relative references.

M69RR \$150.00

SGEN M6809 DISASSEMBLER/SOURCE GENERATOR will produce source code (with symbolic labels) suitable for immediate re-assembly or re-editing. The output source file can be put on tape or disk. A full assembly type output listing with labels and mnemonic instructions can be printed or displayed on your terminal. Large object programs can be segmented into small source files.

M69RS \$50.00

ANNOUNCING TWO NEW M6809 CROSS DEVELOPMENT TOOLS
CROSSBAK—A 8809 TO 8800 CROSS MACROASSEMBLER that runs on your M6809 development system to produce relocatable M6800 object code. Has all features of M69RR (see above). Includes 6800 Linking Loader.

M69CX \$200.00

CROSSGEN—A 6800 OBJECT CODE DISASSEMBLER/SOURCE GENERATOR that runs on your M6809 development system. Has all features of M69RS (see above). An invaluable tool for converting all those 8800 object files over to the M6809.

M69CS \$75.00

All programs are relocatable and come complete with Linking Loader, Programmer's Guide and extensively commented assembly listing. Available on 300 Baud cassette or mini-floppy disk. For disk, specify SSB or FLEX. Source Text input/output is TSC/SSB editor/assembler compatible.

Dealer inquiries welcome.

FLEX is trademark of TSC

Ed Smith's **SOFTWARE WORKS**
P.O. Box 339, Redondo Beach, CA 90277, (213) 373-3380

CT-64

CT-1024

- ★ **FAST! Average Screen Writing Speed: 19K baud**
- ★ Memory Mapped Video Adapter for your CT-64 or CT-1024.
- ★ Just plug it in and go.
- ★ The terminal works like normal until the supplied output routine is used, then the CT-64 or CT-1024 display works at processor speed.
- ★ The J.B.I. Video Board takes 1 main SS50 slot.
- ★ Video Memory can be dip switch selected to any 1K memory slot.
- ★ Combine the J.B.I. Video Board with our up and coming pseudo graphics adapter board and you put new life into the old CT-64.
- ★ The J.B.I. Video Board comes built and is jumper selectable for either the CT-64 or CT-1024.
- ★ If your terminal is CT-1024 we need to know if it is a standard 32 characters per line; or has been modified for 64 characters per line.
- ★ **SHIPPING NOW!**

The J.B.I. Video Board sells for:
\$169.00 with your 21LO2'S
\$179.00 with our 21LO2'S
(8-21LO2'S are required)

We have been a dealer for SWTPC since 1976.

Johnson Micro Computer
2607 E. Charleston
Las Vegas, Nevada 89104
1-702-384-3354

Mastercharge and Visa accepted
Dealer inquiries invited

Software Source BooksTM

Combining detailed descriptions with complete source listings, these books explain the internal operations and algorithms used in Hemenway Associates's popular systems software.

How much would such a complete software resource cost? If you've seen the PAPERBYTE books by Jack Hemenway and Robert Grappel you know how inexpensive they can be. And now you can have the companion volumes to the RA6800ML macro assembler and LINK68 linking loader books.

Remember, these are not just books; they are SoftwareSourceBooks ---- complete Software resources! Order them today; VISA and MasterCard accepted.

CP/68TM OPERATING SYSTEM (\$34.95)

- * PIP Peripheral Interchange
- * Program transfers data between physical devices
- * Wildcard Filenames and Extensions
- * Relocatable anywhere in Memory
- * Extended Instruction set includes 6809-type instructions (PSHX, PULX, etc)
- * Device-independent I/O
- * Random and Sequential Files
- * Fits in less than 8K
- * Chaining and overlaying
- * Single Supervisor Call furnishes all DOS services
- * Easily interfaced to new devices and peripherals
- * Dynamic file allocation

STRUctured BASIC Language (STRUBAL+TM) COMPILER for both business and scientific uses (\$49.95)

- * Variable precision from 4 to 14 digits
- * Structured Programming forms
- * Produces Relocatable and linkable code
- * CO ON and DUMMY sections
- * Extensibility
- * String Handling
- * Full scientific package
- * Data structures with mixed data types

XA6809 Macro Linking Cross Assembler (\$24.95)

- * Runs on any M6800
- * Full Macro facilities
- * COMMON section for the production of ROMable code
- * Conditional Assembly
- * Generates linkable and relocatable code
- * Sorted Symbol table listing
- * Hash-coded Symbol table for speed

=====

Hemenway Associates Inc. 101 Tremont St. Boston MA 02108

Name	Title	Company
Street	City	State Zip

() Check enclosed in the amount of \$.....

() Bill VISA () Bill MasterCard

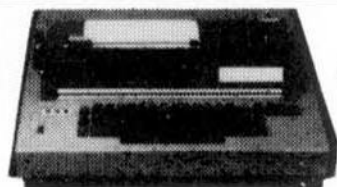
Card No..... Exp. Date.....

Please send the following books:

Add \$0.75 per book to cover postage and handling

SURPLUS ELECTRONICS

ASCII



ASCII

WITH FLEX DRIVERS®
IBM SELECTRIC
BASED I/O TERMINAL
WITH ASCII CONVERSION
INSTALLED \$645.00

- Tape Drives • Cable
- Cassette Drives • Wire
- Power Supplies 12V15A, 12V25A, 5V35A Others. • Displays
- Cabinets • XFMRs • Heat Sinks • Printers • Components

Many other items. SEND \$1.00 FOR CATALOG. REFUNDABLE FIRST ORDER

WORLDWIDE ELECT. INC.
130 Northeastern Blvd.
Nashua, NH 03060
Phone orders accepted using VISA or MC. CALL 603-889-7661

6800/6809 SOFTWARE

ACCOUNTS PAYABLE #1300

Produce financial reports, print checks, special control letter. Reports by vendor number, invoice number, aged and history file. Auto sorting of vendor and invoice files. Plus check and pre check journals. \$600.00

ACCOUNTS RECEIVABLE #1500

Produces financial reports, prints statements, produces reports by customer account number, invoice by customer account number and invoice by invoice number. Print aged report and trial balance. Keeps history file and auto sorting of files. \$600.00

GENERAL LEDGER #100

Program updates to ledger files and also generates reports on payroll, sales, accounts payable, cash and expense statistics. Balance sheet and profit & loss reports. Information can be generated for year end taxes, 941 and W2 forms. \$595.00

INVENTORY I, #600

Inventory for a small company. Produces activity reports for day, month and year. Minimum quantity search. Inventory list by class, by vendor or complete with totals and financial report. \$100.00

INVENTORY II, #700

Produce inventory reports by description or vendor, print activity reports for one day, one month or one year. Quick search by part number, produce total inventory and financial report. (For one store) \$200.00

INVENTORY III, #800

Same as inventory II, #700, but produces reports for eight stores. \$300.00

SHIPPING/RECEIVING

Produce your business financial report, produce reports on sales, accounts receivable and last purchase by customer. Print mailing labels and print customers bills. \$75.00

MAILING LABELS #100

Print mailing labels from your complete file, for a particular city or state. Use one-part mailing labels. \$50.00

MAILING LABELS #400

Same as #100, but also prints labels by names. Use multiple-part labels. \$125.00

BASIC—0935

For those with applications in SWTPC 3.5 BASIC. Runs on 6809 and 6809S. 30% faster and can be used with existing 8800 BASIC Programs. No manual commands and statements same as SWTPC BASIC 3.5. Ideal to keep you going while changing to new BASICs. 5 1/4 or 8 inch 09 Disk, with Renum routine. \$59.95

****Customized programs for your business requirements****
Charge your order to your Visa or MasterCard



Available from Computer Stores or order direct from:
Omnilronics Inc, 1897 Rt. 33, Concord Square,
Hamilton SQ., NJ 08690

6809
6809
6809
6809

COMPUTERWARE

DOS

Same great features as
SSB'S DOS 68.51!!!

MONITOR

Also on the horizon

compatible with new
SWTPC CPU-con-
tains SSB'S disk
boot software

Random BASIC

of course!

EDITOR
ASSEMBLER
TEXT PROCESSOR

write for information

COMPUTERWARE
(714) 436-3512

P.O. Box 668
Encinitas, CA 92024

TRANSITION ENTERPRISES, INC.

We are pleased to announce our entry into the solar energy field. This industry is widely recognized as being in a stage of development similar to the microcomputing industry a few years ago. As we develop new products in this area, we will make them available through our sales representatives. The EX50 (extender board) and CI50 (control Interface) will continue to be available from our dealers, and all correspondence should be addressed to them.

In England:

Sirius Cybernetics, Ltd.
7 Euston Place
Leamington Spa
Warwickshire, England

In Switzerland:

Digicomp AG
Werdstrasse 36
8004 Zurich, Switzerland

In the US:

Disney's Electronics
6153 Fairmount Avenue
Suite 111
San Diego, CA 92120
Floppy Disks, printers & components

SUPER SOFTWARE!

MICROWARE 6800 SOFTWARE IS INNOVATION AND PERFORMANCE

LISP Interpreter

The programming language LISP offers exciting new possibilities for microcomputer applications. A highly interactive interpreter that uses list-type data structures which are simultaneously data and executable instructions. LISP features an unusual structured, recursive function-oriented syntax. Widely used for processing artificial intelligence, education, simulation symbolic and computer-aided design. 6800 LISP requires a minimum of 12K RAM.

Price \$75.00

A/BASIC Compiler

The ever-growing A/BASIC family is threatening old-fashioned assembly language programming in a big way. This BASIC compiler generates pure, fast, efficient 6800 machine language from easy to write BASIC source programs. Uses ultra-fast integer math, extended string functions, boolean operators and real-time operations. Output is ROMable and RUNS WITHOUT ANY RUN-TIME PACKAGE. Disk versions have disk I/O statements and require 12K memory and host DOS. Cassette version runs in 8K and requires RT/68 operating system.

Price: Disk Extended Version 2.1 \$150.00

Cassette Version 1.0 \$65.00

A/BASIC Source Generator

An "add-on" option for A/BASIC Compiler disk versions that adds an extra third pass which generates a full assembly-language output listing AND assembly language source file. Uses original BASIC names and inserts BASIC source lines as comments. SSB and SWTPC Minilisp version available.

Price: \$75.00

A/BASIC Interpreter

Here it is—a super-fast A/BASIC interpreter that is source-compatible with our A/BASIC compiler! Now you can interactively edit, execute and debug A/BASIC programs with the ease of an interpreter—then compile to super-efficient machine language. Also a superb stand-alone applications and control-oriented interpreter. Requires 8K RAM. The cassette version is perfect for Motorola D2 Kits.

Price: \$75.00

RT/68 Real Time Operating System

MIKBUG—compatible ROM that combines an improved monitor/debugger with a powerful multitasking real-time operating system. Supports up to 16 concurrent tasks of 8 priority levels plus real time clock and interrupt control. Thousands in use since 1976 handling all types of applications. Available on 6800 (MIKBUG-type) or 2708 (EPROM-type) ROM. Manual is a classic on 6800 real-time applications and contains a full source program listing.

Price: RT68MX (6800) \$55.00

RT68MXP (2708) \$55.00

6800 CHESS

A challenging chess program for the 6800. Two selectable difficulty levels. Displays formatted chess board on standard terminals. Requires 8K memory. Machine language with A/BASIC source listing.

Price: \$50.00

Our software is available for most popular 6800 systems on cassette or diskette unless otherwise noted. Disk versions available on SSB, SWTPC, or Motorola MDOS. Please specify which you require. Phone orders are welcomed. We accept MASTERCHARGE and VISA. We try to ship orders within 24 hours of receipt. Please call or write if you require additional information or our free catalog. Microware software is available for OEM and custom applications.

MICROWARE
SYSTEMS CORPORATION

P.O. BOX 4665
DES MOINES, IA 50304
(515) 279-8844

'68' MICRO JOURNAL

- ★ The only ALL 6800 Computer Magazine.
- ★ More 6800 material than all the others combined:

MAGAZINE COMPARISON

(2 years)

Monthly Averages
6800 Articles

KB	BYTE	CC	DOBB'S	TOTAL PAGES
7.8	6.4	2.7	2.2	19.1 ea. mo.

Average cost for all four each month: \$5.88
(Based on advertised 1-year subscription price)

'68' cost per month: \$1.21

That's Right! Much, Much More

for About

1/5 the Cost!

1-Year \$14.50 2 Years \$26.00 3 Years \$36.50

OK, PLEASE ENTER MY SUBSCRIPTION

Bill My: Master Charge ☐ — VISA ☐

Card # _____ Exp. Date _____

For ☐ 1-Year ☐ 2 Years ☐ 3 Years

Enclosed: \$ _____

Name _____

Street _____

City _____ State _____ Zip _____

My Computer Is: _____

68 MICRO JOURNAL

3018 Hamill Road

HIXSON, TN 37343

Foreign surface add \$9.50 per year.

Foreign Air Mail add \$29.00 per year.

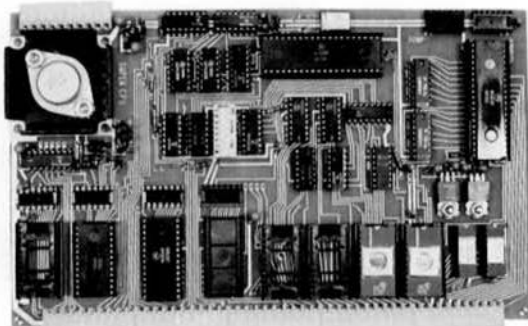


Life subscription \$175.00

NOTE: CANADA & MEXICO ADD \$4.50 per year surface.
New subscriptions require 6-8 weeks processing.

NOW INDUSTRIAL QUALITY AT LOW COST FROM THOMAS INSTRUMENTATION

Industrial system boards are now available separately for OEM, prototyping or hobbyist applications. Shipped from stock, these are the same quality cards used in monitors and machine tool controls designed for GM, LTI, and General Electric. All cards are SS-50 buss compatible and are suitable for dedicated applications. The CPU card and the Video RAM Card may be combined on a TI backplane as a stand-alone micro — ideal for prototypes or hobbyists. ATTENTION OEM's: If you have a control data acquisition, monitoring, or other microprocessor application. Check with TI for more information about custom software design for the TI CPU or any other 6800 series system. TI also has non-SS-50 buss single board 6800 systems.



SS-50 SUPER CPU

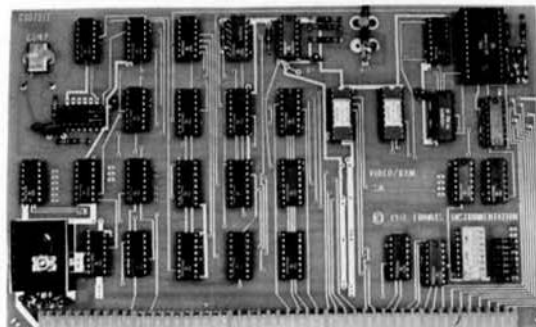
- SS-50 or stand alone computer
- 1K of RAM at \$A000 I/O on board at \$A400 (Relocatable)
- 2K Monitor (Mikbug compatible) in 2708 EPROM
- 2 8 bit parallel ports with 2 control bits and power
- RS-232 ACIA port, 2nd TTL ACIA optional
- 3 16 bit counter/timers (expandable to 6 add 2nd 6840)
- 128 byte RAM at 0000 is jumper selectable
- Battery back-up for 32 bytes of RAM
- Plug back to back with Video RAM for Stand alone micro or customized smart terminal

**ASSEMBLED \$195.00
CARD AND DOCUMENTATION \$49.00**

SS-50 VIDEO RAM

- Fully synchronous operation — No jitter
- 7 by 9 Characters Programmable reverse video
- Full 128 Character ASCII set
- 1K of memory can be mapped to any 1K boundary
- Full documentation includes software (Replaces OUTEE)

**ASSEMBLED \$149.00
CARD, CRYSTAL and DOCUMENTATION \$39.00**



TI SS-50 Wire-Wrap Card	24.00
TI SS-50 Parallel I/O Card	95.00
Card only	35.00
GIMIX 16K Static RAM w/Soft addressing	368.00
GIMIX 16 RAM without Soft addressing	298.00
3, 4, 7 SLOT Backplanes (per slot)	4.00

TI cards available from stock

THOMAS INSTRUMENTATION

168-8th Street, Avalon, N.J. 08202
Phone (609) 967-4280



DEALERS FOR GIMIX SWTPC SSB

CALL FOR DEALER, OEM. AND QUANTITY PRICES

COMPUTERWARE

6800 Specialists Since 1975

Small Business Software

all run on 5" or 8" disk

* NEW *

Accounts Receivable \$149
Same comprehensive features as commercial system without invoicing and statement billing

Accounts Payable \$149
Cash Requirements Projection; detail, summary & YTD reporting; audit trails

Payroll \$149
additional State codes \$50
All Federal & Calif. tax calculations; extensive detail & summary reporting

Inventory \$89.95
Over 1,000 items on a 5" disk

Cash Flow Bookkeeping \$99.95
Detail & Summary reports for user-defined expense ledger

Mailing System \$89.95
User-defined codes for selective sorting

Commercial Systems

Inventory Control \$750
with Order Entry
Bill of Material, Re-order reports; multiple price levels, audit trails; Where-Used & Parts Shortage reports; detail & summary reporting

Accounts Receivable \$600
with Invoicing
Extensive report capabilities including Aged & Overdue reports, statement billing; audit trails; comprehensive invoice entry & posting

* NEW *

Payroll \$500
All Federal tax options; Automatic deduction capabilities; Vacation & sick pay calculations; Extensive detail and summary reporting; complete payroll register; Password protection.
State tax processing \$75.00

Write for complete system descriptions and hardware requirements

Development Tools

RENBAS \$24.95
with source on disk \$34.95
renumbers BASIC programs
resolves line references

XREF \$24.95
with source on disk \$34.95
assembly language program
cross reference

* NEW *

BASREF \$34.95
with source on disk \$44.95
cross reference of BASIC
program line nos & variables

MUST FOR BASIC PROGRAMMER

Random BASIC \$99.95
PRINT USING, EDIT, ON ERROR,
direct random record access

PROM BASIC \$100.00
8K ANSI BASIC, programmable
features of cassette BASIC

Cassette BASIC \$34.95
file handling, 9 digit accuracy,
twice as fast as SWTPC

watch for our

6809 Software

DOS, Monitor,
Random BASIC
etc.

6800 Hardware too!

Smoke Signal Broadcasting,
Centronics, Anadex, NEC,
SOROC, Micro Works,
SWTPC, Leduc
Sanyo, Seals

COMPUTERWARE

P.O. Box 668
1512 Encinitas Blvd.
Encinitas, CA 92024
(714) 436-3512

THE SOURCE

FOR UCSD PASCALTM ON THE 6809

IMMEDIATE DELIVERY FOR SWTPC USERS, 8" or 5¼" DISKETTE
SMOKE SIGNAL BROADCASTING USERS, INQUIRE

- CSI-1 Operating Systems, PASCAL Compiler, Screen Editor, Filer,
Linker, Library, Setup, Binder, Interpreter, BIOS \$250.00
- CSI-2 BASIC Compiler, YALOE, (Line-editor for hard-copy terminals),
Patch, Disassembler, Calculator..... \$100.00
- CSI-3 MACRO Assemblers for 6809, 6800 and other Microprocessors .. \$100.00
- ALL THREE DISKS and MANUAL (SYSTEM) \$419.00

**FREE! UCSD PASCAL USER'S MANUAL PLUS SWTPC
IMPLEMENTATION NOTES WITH PURCHASE OF CSI-1**

68000 Coming, Summer, 1980
68000 Assembler, Spring, 1980

OEM and DEALER INQUIRIES INVITED



1317 CENTRAL AVE. KANSAS CITY, KANSAS 66102

CALL TOLL-FREE (800) 255-4411

Continental U.S.A. Only
(Kansas Residents call 913/371-6136)



"UCSD Pascal" is a registered trademark of The Regents of The University of California

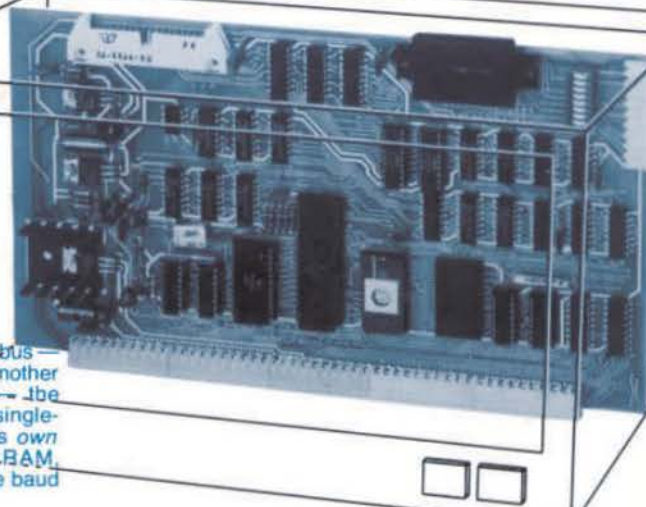


6809 PROCESSING POWER!

The SBC/9. Only \$199.95.



Fully compatible with the SS-50 bus — requiring no modification of the mother board, memory or I/O slots — the SBC/9[™] is also a complete, single-board control computer with its own ROM operating system, RAM, peripheral ports and a full-range baud clock generator.



Make the SBC/9[™] the heart of your computer and put to work the most outstanding microprocessor available, the 6809.

the Mighty 6809

Featuring more addressing modes than any other eight-bit processor, position-independent coding, special 16-bit instructions, efficient argument-passing calls, autoincrement/autodecrement and more, it's no wonder the 6809 has been called the "programmers dream machine."

Moreover, with the 6809 you get a microprocessor whose programs typically use only one-half to two-thirds as much RAM space as required for 6800 systems, and run faster besides.

And to complement the extraordinary 6809, the Percom design team has developed PSYMON[™], an extraordinary 6809 operating system for the SBC/9[™].

PSYMON[™] — Percom SYstem MONitor

Although PSYMON[™] includes a full complement of operating system commands and 15 externally callable

[™] trademark of Percom Data Company, Inc.

utilities, what really sets PSYMON[™] apart is its easy hardware adaptability and command extensibility.

For hardware interfacing, you merely use simple, specific device driver routines that reference a table of parameters called a Device Control Block (DCB). Using this technique, interfacing routines are independent of the operating system.

The basic PSYMON[™] command repertoire may be readily enhanced or modified. When PSYMON[™] first receives system control, it initializes its RAM area, configures its console and then 'looks ahead' for an optional second ROM which you install in a socket provided on the SBC/9[™] card. This ROM contains your own routines that may alter PSYMON[™] pointers and either subtly or radically modify the PSYMON[™] command set. If a second ROM is not installed, control returns immediately to PSYMON[™].

- Provision for multi-address, 8-bit bidirectional parallel I/O data lines for interfacing to devices such as an encoded keyboard.
- A serial interface Reader Control output for a cassette, tape punch/reader or similar device.
- An intelligent data bus: multi-level data bus decoding that allows multiprocessing and bus multiplexing of other bus masters.
- Extended address line capability — accommodating up to 16 megabytes of memory — that does not disable the on-board baud rate clock or require additional hardware in I/O slots.
- On-board devices which are fully decoded so that off-card devices may use adjoining memory space.
- Fully buffered address, control and data lines.

The SBC/9[™], complete with PSYMON[™] in ROM, 1K of RAM and a comprehensive users manual[™] costs just \$199.95.

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

Percom 'peripherals for personal computing'

To place an order or request additional literature call toll-free 1-800-527-1592. For technical information call (214) 272-3421. Orders may be paid by check, money order, C.O.D. or charged to a VISA or Master Charge account. Texas residents must add 5% sales tax.

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE